Enhanced Machine Learning approaches in Text Analysis for Business Intelligence: The appealing story of documents

To my father
Hard as diamond,
Elegant as the morning breeze...

Acknowledgments

Expressing your feelings at the edge of getting your Ph.D. is even harder than doing it. I spent many good and bad days during this time. There were days, where every tiny bit of me wanted to quit, giving their place to happiest moments of joy and satisfaction. But now that I look back, I like to remember only the greatest memories and friendships formed since 4.5 years ago.

I would like to represent my sincerest gratitude to my supervisor, Prof. dr. Koen Vanhoof, who was the only one helping me in promoting to this level. He is the central core of composure in knowledge and his constructive contribution in my PhD is irrevocable.

Moreover, I like to express my love and admiration to the beautiful light of my life, Farnaz. She accompanied me every second passionately and helped me many times with her brilliant ideas. I could not have done this work without her companionship. Only a smile on her face is enough to sweep the sorrow, anxiety and exhaustion from my heart, whether I speak of my concerns or not: She knows everything nonetheless..

Mamma and Baba, I like to thank you truthfully, for all the things I have in my life. You enlightened my way from the first moment. My dear Pourya, Mohsen and Maryam, I like to thank you for your love. I always think about you.

I can not find proper words to state how thankful I am for having invaluable friends in Hasselt who later became part of my family. There is a deep percipience and affection between us, which kept us together in good and bad times during the past 7 years. I regret that I did not have the chance to be close anymore since we moved to Germany in 2016. I like to say thank you to Ali, Elham, little Elena, Majid, Seddighe, little Radin, Mehran, Fereshteh, Amir, Negar, Farnoosh and Roohi.

I represent my respect and appreciation to my Ph.D. jury members for the constructive feedback they provided. Also I would like to thank my colleagues who shaped my academic character: Frank, Hanne, Niels, Gonzalo, Gert, Toon and Marijke, Thank you all. Special thanks goes to my partner Mathijs for his patience whenever I came over with many questions and for all research we did together. My brothers Hamzah and Ahmad: I will always remember you with your eastern kindness and sympathy. Lia, you are a great friend and always merciful to me. Thank you. Mehrnush, you are always helpful to me with your energizing words. Kudos! Also no matter where I will be from now on, I will never forget Mr. Wouter Faes, who is a great compassionate friend to everybody, specially to foreign students. His visits to our office always brought joy!

Finally, I would like to thank Prof. dr. Elpiniki Papageorgiou, dr. Bastian Steinert and dr. Guy De Pauw for their support and suggestions.

Aziz Yarahmadi
Hamburg, September 2018

Contents

List of Tables

List of Figures

Preface

We live in the era of ruling business information systems. Many Business Intelligence (BI) applications are developed daily to improve the quality of services provided by businesses. The spread of big data phenomena corresponds to substantial data sources and data warehousing techniques in order to handle data querying. Raw data on the other hand, requires proper analysis to reveal patterns. Knowledge could be extracted from raw data using BI techniques. While most BI techniques tend to deal with numbers as the data type, we have focused on information in the form of text.

Documents as the main source of text data, are being interpreted from different points of view. Researchers are interested in different aspects of information presented in documents. As an example, *Semantic Analysis* investigates roles and meanings of expressions, clauses and words beyond their superficial entity in documents. Market analysts on the other hand, are employing *Sentiment Analysis* to determine the polarity of opinions expressed in the form of reviews and comments regarding a certain subject. There are *Text Classification* and *Knowledge Discovery* as well, along with other minor applications.

Text mining applications in this matter, enrich decision making systems. With respect to quality of text data, informative features are extracted using text mining tools, which further empower deficit tasks and role players. On the other hand, huge volume of text data exists in business data warehousing systems, that needs to be reviewed in every specific case. Such a procedure is time and resource consuming. Text mining applications are capable of automating this procedure by going through a document, extracting meaningful sentences, interpreting valued knowledge and finally store and maintain them in form of updated decision making routines.

In this thesis, we have studied the concept of text analysis. This book represents the detailed discussion of prerequisites of a text mining application including know-how on different types of text, preparing the text for the means of analysis, the techniques to extract and analyze text data and issues which an application is aiming to address. In other words, we have presented an explicit literature review of text mining applications as the infrastructure of the second main part of this study. In the second part, we have introduced two novel text mining applications. These comprehensive applications explore less discovered fields on text analysis.

First, we have discussed the verge of data mining pattern extraction in text mining using databases with descriptions expressed in a combination of two languages. This work, provides a classification schema for massive databases of commodity deliveries. The schema presented in this study is able to categorize any number of products, based on their descriptions introduced in both Arabic and English. The novel approach in this work is flexible to handle even more than two languages and in this way differentiates itself from other text mining applications employed by data mining. The second application conducts a series of experiments in order to prove a novel feature extraction framework for regulation based documents. The experiments prove that documents composed of rules can be classified using the consistency of patterns which exist in a rule structure. We introduced the use of N-grams generated from part-of-speech tags, which reinforces the solo use of word as features. In this way, the text classifiers were able to classify rules with a high precision, resulting in an infrastructure for decision model extraction platform from regulations.

In both studies in this thesis, we have employed text mining principles in order to exhibit capabilities of text data pattern recognition as a part of business intelligence applications in daily tasks. Our frameworks are addressing two of the most popular data analysis issues: product classification and rule classification as the basis of decision model extraction. The results show that these applications possess the competitive advantage over similar applications. The novelty of applications makes them proper for broader application in other languages on one hand, and more powerful and less resource demanding with an optimized feature extraction schema on the other hand.

Chapter 1

Introduction

Chapter 1 introduces the fundamental representation of text mining as a concept. Initial impressions, primary applications and further developments in this field are respectively discussed. The background of challenges we face and the research motivation continues the introduction chapter. Next, the thesis approach is briefly discussed in research methodology section. The final section provides an overview of the thesis structure and organization.

1.1 Background

With the emerge of *Business Intelligence* as an answer to overwhelming growth of platforms generating raw data in business, we observe derived pattern detection infrastructures for different types of data day-by-day. Luhn [1] presented his novel theory of Business Intelligence, defined as a set of comprehensive interrelationships of discovered facts over a business that aim to facilitate the business toward a desired goal. Since then, researchers have developed a vast number of applications in data analysis, providing multiple units in a business with real-time access to meaningful knowledge extracted out of data. *Big Data* technologies, initially introduced by Mashey [2], correspond to a variety of methodologies, software and problem-solving frameworks which focus on intensive databases and conduct experiments and routines according to data types available. Data analysts apply these methodologies and frameworks on massive semi-structured or unstructured datasets to investigate whether a business is following road maps towards corporate goals.

Text data has recently gained reputation among data analysts exploring Big Data context. With the spread of Social Networking as the most popular means of communication in World Wide Web, establishments are utilizing data warehousing and processing power of web-based applications extensively. Moreover, text data generated daily is playing a major role in data transmission means. From an information retrieval point of view, text data is composed of two formats: Representative and Collective. Representative text data expresses facts regarding a subject, while the collective text data gathers objective facts. Known examples of representative text inlcude news headlines and instruction manuals, while surveys and questions are famous instances of collective text. Representative text provides information access to users [3]. In this regard, various services are implemented, presenting ideas and facts on a subject that matters to users. Books, newspapers, financial documents, contracts etc. are all means of knowledge sharing nowadays. Study shows that at least 80% of data in a company is in text form [4]. However, knowledge in the form of text data is not always evident, or in many cases, trends and major patterns exist along simple text which need to be revealed. Companies have recently recognized the added value of text analysis, appeared in marketing, technical planning, legal inspection and financing and investment opportunities. Examples include the assessment of a commercial product penetration ratio based on social media users comments, project requirements documentation manuals, similar case scenario handling evidence on the web in legal affairs, etc.

In such a perspective, the concept of *Text Mining* has brought analysis capabilities to text data science. Gupta and Lehal [5] define text mining as the process of extracting profitable and non-trivial knowledge from semi-structured or unstructured text. In this matter, some researchers identify text mining as a sub-concept of *Data Mining*. While text mining inherits fundamental concepts and approaches from data mining, the ability of handling unstructured text data stands in capabilities of text mining [5]. Primary applications of text mining were governed by national security issues and critical life-sciences [6]. Furthermore, approaches in text mining embrace three major categories of applications: *Information Retrieval* (IR), *Information Extraction* and *Natural Language Processing* (NLP) [7].

Information Retrieval is the ground concept for web search functionality and basically provides collections of documents which contain information on the data query. Applications in this field vary from internet search engines and keyword follow up to image data extraction [8]. A *Language Model* approach toward information retrieval was introduced by Ponte and Croft [9], where authors suggest that a language model should be created for each document in the database. Consequently ranking the results is possible by assignment of a probability to each query text according to the language model of a document.

On the other hand, Information Extraction is the general procedure of providing structure for unstructured databases. In contrast with Information Retrieval, which is based on returning meaningful results to a human-understandable query, Information Extraction satisfies the prerequisite of structuring for machine readable data. In [10], the author has presented the general rules of information extraction from semi-structured and free text data. Natural Language Processing is the set of theories, applications and techniques that empower machines to understand and interpret human languages [11]. Considering the diversity and complexity of human languages from structural and contextual points of view, substantial number of applications are developed in this field. Enriching *Machine Learning* techniques in this regard, enables researchers to extract meaningful information from bulky document datasets. Machine learning paradigm predicates the ability of computers to learn the educative attributes of a phenomena, provided the fact that a set of trained samples are accessible. Hence, categorizing and clustering text, detecting the subject of concept popularly known as *Named Entity Recognition (NER)*, recognizing the polarity of a statement-known as *Sentiment Analysis*-and determining treatment of words according to document context [12]-known as *Semantic Analysis*-are all principles of machine learning in natural language processing. Furthermore, a couple of application fields in this regard, are briefly introduced.

In [13] the author has defined the task of automatic text classification. Assuming a certain pair of $< d_j, c_i > \in D \times C$, where D is a collection of documents and $C = \{c_1, c_2, ...c_n\}$ represents pre-determined classes, a target function which is also called the *Classifier* $\phi : D \times C \rightarrow \{F, T\}$ assigns a Value of T to d_j, to confirm it belongs to class c_i. A value of F defines that class c_i does not include d_j. *Document organization* and *text filtering* are two specific applications in this regard. Ritter et al. [14] has promoted the performance of NLP pipelines through *chunking*, as an answer to the overwhelming NER in tweets from Twitter. Twitter data, as one of favorable microblogging social networks, contains noise mixed sarcastic definitions, which reduces learning ability of machine learning applications to determine specific subjects. Liu [15] has stated the problem that sentiment analysis is addressing. Statements containing both positive and negative words or expressions, require a weighting measure for each word according to a dictionary of terms and their relative weights. In [16], the authors have proposed a novel methodology that realizes meanings of word by creating weighted vectors inherited from Wikipedia articles.

Nowadays focus on principles of machine learning capabilities is the research and industry interest for many specialists. Growth in use of machine learning techniques has encouraged text mining researchers as well, to expand similar applications for further use cases and develop new applications. With respect to definition of 3 main categories of text mining applications introduced above, experiments presented in this study and the type of facilities these experiments provide in Business Intelligence are situated on a combination level of Information Extraction and Natural Language Processing. In other words, Information Extraction provides the infrastructure needed in text data structuring, so that Natural Language Processing can extract the latent patterns which lead to automatic classification of text documents.

1.2 Challenges

Although the use of text analysis applications is increasing drastically during recent years, the quality and organization of text data has remained a challenging empirical issue. Text data is not presentable to the machine in its raw format. As a result, including documents as data in databases depends on language, length and relationship of text data with other data attributes. Independent and technically proper databases of documents are mostly created and maintained as a criterion to evaluate other applications. Many of these databases have been as well established and developed as part of academic projects. *20 newsgroups* [17] and *Penn Treebank Corpus* [18] are known examples of such databases. Degree of compliance with new extensive data analysis projects considerably depends on quality of data. Furthermore, text data quality is directly correlated with the source of data. In other words, documents generated by government officials, newspapers, research articles, manuals and regulatory organizations follow a high degree of formality and precision regarding the use of grammar and words, compared to text data in personal blogs, social media and informal conversations in letters and instant message platforms. Social media application programming interfaces (API) support data extraction for analysis purposes. In this way, a query containing specific words will return every document in which the word exists. Hence, data collection includes several languages, formal and informal writing styles, noise in the form of hyperlinks, emojis and slang idioms. As long as researchers focus on classic formalities with the help from reference collection of documents, pattern recognition chances decrease while dealing with such data. Therefore, two major principles of *data source selection-or proper query-* and *data pre-processing and cleansing* are of great importance and require more investigation in different text analysis tasks and context.

Data pre-processing has proved to improve performance accuracy of a *Support Vector Machine* classifier considerably in [19]. *Stemming*, which means to convert all forms of a word to its root form, shows improvement in the classification performance in [20]. Text data pre-processing demands around 80% of the time span in a text analysis project. Despite considerable growth in the scope and abilities of preparing text data to be a proper machine learning input, anomalies in the form of text data including terms or phrases from different languages would require further processing strategies. Due to different reasons such as simplifying the text in local regions, text might be presented in a collection of different languages. In text analysis, it is necessary to gather the highest level of details on the nature of text. As a result, the challenge is to benefit all terms from different languages.

Additionally, a major number of text mining applications only concentrate on words used in a document, in order to perform any text analysis task, from *Email filtering* to *Employer Branding*. Existence or absence of words corresponds with classes and weights, and because of this reason, researchers are interested in transforming words into numerical analysis features which are presentable to the machine as input. Evidence in [21] expresses the fact that the order of words in documents are of minor importance. Each word w_i is represented as a feature $f_{w_i,t}$, where, t shows the number of times w_i appears in a document. Pang and Lee [22], have examined N-grams of words in a sentiment analysis task using Naive Bayes, support vector machines and maximum entropy classifiers. Considering words as learning features has also proved to create high dimensional sparse matrices from these learning features while training classifiers, due to the fact that the number of context-specific words grows rapidly in documents, and small training datasets do not provide added value to solve this case [23]. As a result, improving methods for *dimension reduction* leads researchers to *Term Frequency (TF)*, *Information Gain (IG)* [24], *Mutual Information (MI)* [25], χ^2*-test* [26] and *Term Strength (TS)* [27]. The ultimate goal in dimension reduction-or feature reduction-is to reduce the non-informative selection of terms, which decreases the time and processing power needed to train predictive model. Although several studies explored reducing the number of learning features for relatively small datasets, less attention has been paid to large scale text classification-e.g. very long documents with hundreds of thousand of terms-. Moreover, selection of the proper learning feature reduction-i.e. choosing only the parts of text that can help find similar text types-remains a challenging issue, as it considerably depends on the type of text structure and the analysis goals.

The final challenge originates from the fact that very few studies focus on semantics in terms of linguistic structures, parsing variations and organization of terms by their grammatical roles in documents, to be used as learning attributes in defining characteristics of text, such as a class. For instance, no study has tried to generalize a stable sentence structure for documents from the same family. Determining a number of grammatical structures can potentially identify relevant classes and similar documents respectively. Resolving such a challenge can result in applications in categorizing and filtering scopes which are very interesting to the business world.

In the next section, we have framed the research motivation and steps taken in this study to deal with these challenges.

1.3 Research motivation

A couple of motivations reached us to a place in generalizing the challenges expressed before. We believe that resolving these issues using new research methodologies and techniques is broadly requested in business intelligence and can lead to better pattern extraction for text data analysis tasks.

The initial research motivation in this study originated from a request by a parcel delivery company in Middle East. Having provided a dataset of parcels delivered to private customers, the company expressed a need for a classification platform of parcels based on their content. This brought the first field test. The goal in this project was to classify the items based on a data attribute, introducing a description of content of the parcel and to assign a tag from a set of 70 predefined groups to each parcel. In other words, by referencing ground theories and methodologies of text classification, we embedded text classification influence to a common data mining solution. The company was planning to monetize the output knowledge as marketing insight, in which vendors and goods merchants are highly interested. One specific monetizing potential of such an insight is the identification of the company as a leading delivery partner of certain consumer goods and major online merchandise holdings such as Souq[1]. Having stated the need for pre-processing indistinct data attributes, the dataset provided over a million records of delivery items of consumer goods. The challenge rises from a dual language script used in descriptions and the vast scope of words there, which leads to a drastic growth in number of features for classification. In order to overcome such complications, we first enter the diversified realm of supervised text classification. The establishments in introducing text data type as an input to analysis platforms and preparing dual language documents have been brought in chapter 3. Then, the implementation of the solution for parcel delivery company is presented in chapter 4.

The second phase of this study, covers another text mining application requested during an external research stay in Germany. A certain legal affairs department representative expressed the need for an infrastructure that facilitates the automation of regulatory compliance checks. Generally, rule-based documents contain one or more types of rules which correspond to relevant information and situations and provide the know-how on flow of business. Such an automation procedure can be viewed as the fundamental properties detection in extracting a *decision model.* Transformation of a rule into a decision model provides a maintainable rule management and application system. In this regard, I have addressed the possibility of classification feature selection based on the reoccurring document structure as a novel approach in supervised text classification. State-of-art research in document classification though, has mainly focused on consideration of words and the frequency of distinctive terms as the classification feature. As stated before, training classifiers for rules according to stable grammar structure of the rule, can potentially lead to better classification performance, compared to employing words as features. The dilemma over effective feature selection in regulatory-based content and relevant theories are introduced in chapter 5. The design procedure, data structure and samples, feature preparation and results verification on sample documents have been presented in chapter 6, along with highly satisfying results in text classification capabilities of the experiment and methodology.

With a clear view over the global scope of the research and motivations introduced, in this study we aim to overcome the following challenges:

1. How to treat different text data structures with respect to different pre-processing techniques, vast set of specific terminology, the effect of previous knowledge on the task precision and different text analysis tools.

2. How can text analysis approaches and techniques be modified during classification of documents containing uninformative characteristics in terms of class selection and words from two or more languages, in order to bring added value in problem solving tasks in commercial data mining?

3. How can a classification schema for documents including rules be designed in order to act as the ground infrastructure of an automatic decision model

[1] https://uae.souq.com/ae-en/

extraction platform from documents?

Further, the research methodology in confronting the aforementioned challenges is presented.

1.4 Research methodology

To address the challenges presented in the previous section, one needs to perceive the big image in understanding the fundamentals of text classification. Existence of human supervision, in the form of providing classified data samples to be used as a guide through introduction of the learning model is the game changing element. As a result, different approaches, algorithms and techniques can be determined in text classification with or without previously annotated data. We begin this study by a close look on the concept of text classification and two two major approaches of *Supervised* and *Unsupervised* text classification, which originate from principles of machine learning. A literature review on these approaches, relevant terminology in different elements of these approaches along with analysis of different text data structures in composing a text classification model is undertaken. The goal in this part is to build an image towards selecting a proper approach in specific text classification problems.

To address the second challenge, we basically aim to provide a text classification framework to be used in commercial machine learning problems which enhances categorization of services, products, customers, etc. Complexities arise when the dataset to be classified includes words or phrases from two or more languages in any of data attributes. Furthermore, text data in this regard might include many irrelevant words to the class and unwanted information in the form of characters and numbers, which impose higher necessary processing power and more resources. It is in our interest to keep only the information which can be turned into knowledge in class prediction and transform the data into a unified set of words from one specific language to facilitate the classification task. In realization of this goal, we initially propose a normalizing function to convert all non-English terms into English. Furthermore, a set of pre-processing tasks prune unwanted data and prepares text data to be converted into appropriate machine inputs. A selection function is implemented consequently based on the weights each term is dedicated according to its relativity to the context and the number of times it occurs to choose only relevant information for model training. Next, using previously classified data samples, different classification models are trained based on a number of classification algorithms which previously have returned considerable results in text classification tasks. Finally, a section of the data that has been kept for evaluating the classification models will be given to the machine. Hence, a classification schema is shaped for further similar data samples using the most accurate model. Specific characteristics in supervised text classification includes the previously known classes and data samples assigned to each class. To evaluate the validity of the approach, an experiment is designed based on a real case study.

The goal in this study is to classify thousands of rows of items, delivered by a parcel delivery company. The experiment design and steps taken in implementing a supervised text classification schema for this data are presented in chapter 4.

The goal in solving the final challenge is to represent yet another application of text classification in business intelligence. As transforming decision making approaches from documents into executable decision models is an interesting topic for the experts, solutions to such a problem are highly valued. Determination of the structure of documents including rules is of great importance by out theory. The first step in generating a decision model is understanding different types of rules and their structure. In this regard, we focus on providing a classification system for different types of rules in this study. As we again look at the problem from the supervised machine learning point of view, access to previously classified data and specific classes are required. We have proposed a novel approach in which the various grammatical structures of a limited number of classes of rules are identified. These grammatical structures are defined using grammatical roles of words. We propose that a combination of these grammatical structures along with the words themselves can provide the highest level of class prediction capabilities. Preparing the text for appropriate machine inputs and cleansing text data are executed in the next step using above methodology. Next, the classified data from before is used to test the model. Using stratified cross validation to ensure the existence of all classes in the training phase, we run 10 rounds of training and testing the models which are implemented based on a number of classification algorithms. Such a methodology results in a highly precise document classification framework using 4 classes of rule types. In realization of our proposed methodology, an experiment is designed to test the capabilities of the classification schema on a set of rule-based documents.

In the next section, organization of chapters in the rest of this thesis is presented to provide a road map in achieving the goals expressed above.

1.5 Thesis organization

In realization of the research methodology, chapter 2 presents the ground knowledge relative to the general concept of text data classification. Respectively, the importance of text data quality and data cleansing and pre-processing required for each type of text is determined

Chapter 3 starts with a background on the rise of supervised text classification from data mining. The need for a transformation of text data into numerical features is further discussed and state-of-the-art feature extraction models are explored. With respect to text pre-processing and cleansing, a brief introduction to NLP capabilities of Python programming language is presented as well. *WordNet* technology, as a network of words, the word hierarchy within and the role it plays toward document classification is outlined.

To employ the concept of supervised text classification, chapter 4 presents the first experiment designed in addressing the second main challenge in research motivation. This chapter reviews the text analysis facilities required by a parcel delivery company and we prove that the use of dictionaries in mapping terms using a network of words is efficient in text classification. This study is published as a scientific paper in an academic journal and is accessible by public [28].

Chapter 5 deals with *Vector Space Model* theory and builds requisite background knowledge for the experiments conducted and discussed in chapter 6. Moreover, the use of N-grams of characters in text classification is elaborated.

Chapter 6 provides the results of our second application, aimed to classify rule-based documents. A novel approach towards text classification feature extraction model using *N-grams of Part of Speech tags* combined with their respective terms is employed and we have proved this framework to be effective and highly precise in regulatory compliance check document classification.

Finally, chapter 7 outlines the conclusion of this thesis and further studies which can be triggered by these experiments. Further work over *Decision Modeling* from raw text, is presented as a very specific and interesting research topic.

Chapter 2

Document Classification Fundamentals

In chapter 2, a fundamental study covering the ground concepts of text classification that are utilized in next chapters is presented. Moreover, an overview of the context-based terminology which is frequently used in the next chapters continues this chapter. Further, we review the concept of a text classification *feature* and language model representation along with introduction of Part-of-Speech tagging. We determine the significance of text data preparation and hence, review the pre-processing techniques in data cleansing.

2.1 Text Classification

Among the broad range of text analysis applications, categorizing similar documents are one of the most in demand groups. Due to generation of text data in business environment by a fast pace, computing facilities are required daily to group documents with close characteristics. In this regard, many use cases can be figured out. Sales departments for example, demand a classification framework to divide presentation material, sales invoice, proforma invoice and payment receipts automatically. In marketing tasks, customer segments are identified based on the customer reviews or discussions in consumer product forums. Previous known applications such as spam email filtering would come to mind as a use case of text classification as well. To classify documents, researchers have taken different approaches. In [29], the authors have proposed a hierarchy of classes, to shrink the class diversity and to improve the classification task. Xiang et al., suggest a character-level convolutional network for a precise document classification [30]. Semantic kernels derived from Wikipedia in [31] are employed to strengthen the background knowledge and enhance document classification. These and many more other studies propose various document representations that are used to train classification algorithms. In order to perceive an appropriate image of text classification (TC), one needs to understand the major general frames of techniques and approaches towards TC. Among all, *Machine Learning (ML)* has proved to be effective in automatic class prediction of documents compared to manual approaches [32].

From the Machine Learning point of view, an automatic document classification system is either built on a set of documents with known class labels or requires to detect the classes-or clusters-without any prior knowledge over the class characteristics. That being said, the choice of approach selection among these two, hugely depends on the nature of TC task and the amount of information at hand. News classification for example, enjoys substantial previously categorized topics with hundreds of thousands of articles-documents-that facilitate taking the first approach. Detecting the subjectivity of news readers comments in news websites though, falls in the field of *Sentiment Analysis*. Due to the fact that the language in this case is informal and contains irony and sarcasm, lack of precedent knowledge would encourage the scientists to follow the second approach. In ML, the first approach is known as *Supervised learning* and the second is known as *Unsupervised Learning* to stress the human supervision or lack of it during the analysis.

Unsupervised TC tends to group documents based on their similarity and recently, there are two major methodologies in performing such a task. One common strategy is to compare the similarity of two documents represented by their terms and second is to train a clustering algorithm to position each document in their relative cluster [33]. The latter, due to the extension capabilities to multiple context and domains is more in demand but requires considerable time and processing resources. The reason lies in the fact that with the lack of previously annotated data, multiple points of focus-or learning features-would come to mind to determine a relative cluster for a document. Consider the case of an electronic product review by users where each customer expresses various viewpoints from the product features and quality to the general attitude or feeling towards the product. An example is followed regarding the gaming console Xbox one S:

"I ended up purchasing this because the cost was slightly cheaper due to it being refurbished, but I like refurbished, that means it is fixed up to be like new. Yet, the console came in a cracked custom box " (2.1)

The user has expressed satisfying cheap price and a tedious cracked box, but generally is happy to buy a refurbished product. Now, the question is whether the classification algorithm focuses on the dissatisfying feature or the overall content attitude from the purchase.

Supervised TC on the other hand, is provided with a set of known classes and pre-labeled samples of text documents. In product categorization for example, merchants group new products based on a set of product classes and records of products which are already sold. Each product is categorized based either on size, price, customer type or any other criteria demanded. For each of these attributes, records of products are categorized previously. Since we are going to employ NLP techniques to extract knowledge in the form of document classes, it is necessary to follow the NLP guidelines and approaches in representation of potential supervised and unsupervised text classification approaches. On the other hand, since text data is not a proper input for computer programs, documents are to be converted to numerical representations. Additionally, each classification model needs to be trained based on a distinctive attribute of text data, so it can be able to calculate the degree of similarity among documents. In the next chapter, such a characterizing phenomenon, known as a *text classification feature* is discussed.

2.2 Machine Learning text features: Selection vs. Extraction

In machine learning problems, a learning *feature* is an attribute of the entity which is being analyzed and determination of features facilitate the learning procedure. With simpler words, a classification algorithm learns to categorize data samples based on certain data attributes. In categorizing cars based on the amount of air pollution they cause for instance, consider the case of inspection of two cars. If car A engine emits 4.7 tons of carbon dioxide per year, in order to find out if car B stands close to car A in terms of carbon dioxide emission, one needs to find the amount of emission for all cars in the category in which the car A stands and define other categories with different ranges of emissions. In other words, the learning feature to categorize car B in a proper class is the amount of emission of carbon dioxide.

From the ML point of view, automatic text classification requires the definition of specific features for categorizing documents as well. In text classification and at the most frequent use level, words -sometimes known as *terms*, and are extensively used for one another-and phrases are considered as features traditionally [27] and in this regard, *Feature Selection* is the process of choosing a manageable subset of terms that is sufficient and effective in describing the subject of analysis [34]. With the amount of text data growing and as far as documents include tens of thousands of words, the number of features increases progressively. Yet, many term features propose near zero information gain and do not play any role in describing a document. For instance, terms such as *and, so, for, this*, etc. occur in different types and almost every kind of documents and in the context of text classification, they provide no information on the document genre. Hence, text data structure is known to be of high dimensions, since in text data representation, each document is identified in a multi-dimension space, with each feature acting as a certain dimension. Reducing the number of uninformative features or *dimension reduction* has proved to require less processing power and increase the performance in effective prediction [34].

Dimension reduction provides a knowledge expansion environment by utilizing two approaches. The first one is selecting a set of distinctive and informative features from the original feature set, like the complete set of terms occurring in a document, that would result in a higher class prediction precision, along with less computing and memory assignment. As stated before, such a task is known as feature selection and is supported by many methodologies such as *frequency of terms*, *information gain* [21] and *mutual information* [35]. In this scope, based on existence of class information-for instance terms belonging to each class-feature selection can be of supervised or unsupervised type [36]. When a class label is available, a measure calculates the distance or relationship of the current feature with the reference feature, resulting in a ranking of features, which then is used to sort out the most relevant features. Methods such as *information gain (IG)* or χ^2 belong to this category and are explicitly introduced in the next chapter. Unsupervised feature selection on the other hand, is not benefiting the class information and hence, is not feasible to be used when the text is to be grouped into clusters of relevant documents. Methodologies such as *term frequency*-to calculate and rank the number of occurrences of terms in each document-and *term variance*-to lower the effect of low frequent terms by calculating the variance of all terms in the dataset-are developed to address unsupervised feature selection problem. Yet, without removing a defined set of terms, unsupervised feature selection is not an efficient approach [36].

Moreover, *Feature Extraction* is the second approach in providing more insight over the efficient TC task. Feature extraction selects a subset of whole dataset of features, but as well transforms the features into numerical representations that are appropriate machine inputs. This means that a set of features-which can include strings of characters, terms or bigger languague units-might be selected that represent others based on the transformation function. In other words, feature selection supports the prerequisite preparatory task for feature extraction [37]. Among all, two approaches of *word2vec* [38] and *term frequency-inverse document frequency (Tf-Idf)* [39] are widely used to transform textual features-which can be characters, terms or set of sequential terms, etc-into numerical features. Both of these approaches, embed documents in a *Vector Space Model*, with each document represented as a vector. Tf-Idf is broadly introduced in chapter 3, as we have used this approach through this study extensively.

Yet, to determine the impact of feature extraction, one needs to perceive its ground infrastructure. In this respect, the *token* which is the most basic language model unit of analysis should be identified. If we define the *corpus* as an entire set of documents, in each text analysis task, a corpus can be of different size. A corpus can be the size of a paragraph with a limited number of sentences as well. Hence, in such a corpus, each document is a sentence of the paragraph and respectively, a token is identified based on the value of information it provides. This means that experts define the length of the token, which is part of the corpus, based on their goal and text analysis facilities. In other words, a token can be a character, a set of characters, terms, a set of terms, a sentence or the even bigger language model units. To perform any text analysis task, one should first define at which level of tokens from a language model they want to work. For instance, at the most basic level in this study, a token is all of the terms occurring in a document. Higher levels include word combinations and sentences. Splitting text documents into well defined tokens-which is known as *tokenization*-makes them appropriate as inputs to text analysis environments. With a clear image over the structure of a token, further feature extraction steps can be initiated. Technically, the concept of text quality prioritizes the task of document pre-processing in appropriate feature extraction. In other words, to extract informative tokens and proper features respectively, the quality of text should be enhanced. In the next section, we introduce the concept of text cleansing and tasks which result in extraction of class representative features.

2.3 Data quality and pre-processing

The previous section shed light on the role of class distinctive features and their potential in facilitating categorization task. In addition, many characters or terms exist in each certain document, that originate from the socio-demographics of the population where the text is generated. Financial documents for example, contain numbers and regulation characters or text from social networks used by young generation contains special characters like exclamation marks etc. Although treating such terms or characters is in the scope of some text mining applications like opinion mining-or sentiment analysis, which employs implicit knowledge in such characters such as emojis-in many cases these terms provide no knowledge over the nature of document and need to be removed to lower the need for processing power and increasing efficiency. Pre-processing is the complete set of techniques and approaches conducted in order to prune and clean raw text data and to restructure text entities in a model input oriented procedure. To transfer the term features into the machine readable space, feature extraction depend on a couple of pre-processing techniques as well. In this regard, pre-processing defines a series of steps to clean data from a set of objects and to transform textual features. With the spread of natural language processing, more and more programming languages are equipped with implementation of text mining techniques. Hence, we present a set of text cleansing techniques and transformation functions with related methods in Python programming language.

2.3.1 Parsing and tokenization

Raw text is passed through a pipeline of cleansing and preparing tasks. The first step in text preparation is *text parsing.* The role of a text parser can be imagined as the text inspection by a human user. The act of parsing is a conformity check for structure and language grammar. In other words a parser detects if the data in use is text and confirms the basic infrastructure of a certain language in text data.

Furthermore, a tokenizer splits the text into proper constitutive entities of choice, the tokens. These units of interest as we said, can be of the form strings, words and sentences. Among all, researchers are more interested in words, word combinations and sentences, as with lowering the size of strings of text, information transition decreases. The basic concept behind converting text into tokens- or tokenization- is to prepare the units of analysis. Two highly in-use approaches to detect term tokens include the use of *white space* and *regular expressions*. In the first approach a tokenizer detects white space among words and entities on either side of the white space are selected as words. In tokenization into sentences, use of *full stop* counts as determination of a sentence instead of white space. In the second approach, regular expressions, which define a set of search patterns are employed to detect tokens. According to [40], a regular expression denotes a set of strings (i.e. a language) or string pairs (i.e. a relation). It can be compiled into a finite-state network that compactly encodes the corresponding language or relation that may be infinite.

For a clear view over parsing and tokenization, consider the following short passage of text from CNN:

"I think it is impossible to stop the trade," he said. "The world needs trade. If the trade stops, the wars start."

(2.2)

A text parser detects that the data input is of text category and specifies the English language. Moreover, tokenization into words and sentences takes place using white space and full stop sign respectively. Yet, defining a regular expression to split text into words can be done as follows in Python:

```
re.findall("[A-Z]{2,}(?![a-z])|[A-Z][a-z]+(?=[A-Z])|[\'\w\-]+",s)
```

(2.3)

re is the specified library of functions for regular expressions in Python and *findall* method in the upper regular expression finds all patterns of strings which in this case can start by upper case alphabet letters, are composed of at least two letters, and continued with any lower case letters. Additionally it can be composed of a uppercase letter followed by a lowercase letter plus a look-ahead group of zero or more letters and finally is formed in a word format, according to the method instructions. In addition to regular expressions, Python provides *split* and *join* methods along with *encode* and *decode* methods to detect non-utf8 encoding for non-ASCII character removal.

In this field, *parse trees* are as well supportive learning facilities that represent a machine's ability to figure out either language model dependencies or phrase structures [1]. To understand dependencies, consider the following sentence:

"I saw some birds flying over the mountains"

(2.4)

This is an ambiguous sentence as the human interpreter is not assured whether the act of flying is related to the birds or it is the status of the speaker while he/she was seeing the birds. As a result, a parse tree can be formed in two main formats which are *Typed dependency* and *Parse tree of structure*. The first format is derived from Stanford parser [2]. *Stanford typed dependencies* which indicate the structural relationship between sentence entities are employed in this first type. Following Stanford Typed Dependencies manual, *(nsubj)* determines the syntactic subject of a clause. *(dobj)* is the abbreviation for direct object and defines the object of a verb. As the tree in figure 2.1 suggests, *birds* is *(dobj)* for the verb phrase *saw*, and *I* is the *(nsubj)* for the verb phrase *saw* respectively. Some noun phrase determiners are indicated as well with *(det)*. Additionally, a noun phrase modifier *(nmod)* modifies the meaning of noun phrases *flying* and *mountains*. In fact, Stanford parser here enhances these two nouns by providing explanations. Yet, the distinctive feature is where *flying* acts as the *adjectival modifier (amod)* for the word *birds*, describing birds act while the writer was *seeing the birds*. This is the first grammatical viewpoint. Refer to figure 2.1 for a more clear image in this case.

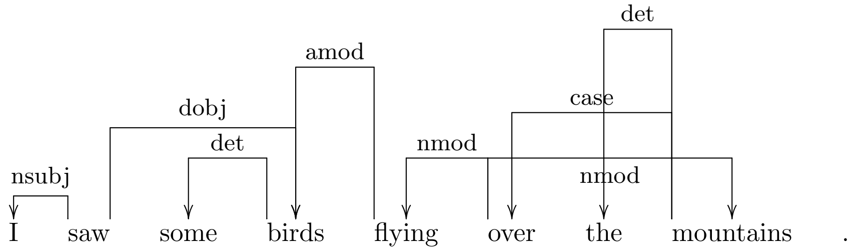


Figure 2.1: Typed dependency tree of the sample sentence: first point of view

From the second grammatical viewpoint, a second form of typed dependency tree can be derived from the sentence, if we determine the NP-noun phrase-"*flying over the mountains*" as the adverbial clause modifier *(advcl)* to the root section of the sentence. This means, the whole act of *flying over the mountains* serves as a definitive time when *seeing the birds* happened. Hence, we can elaborate these

[1] NLP Programming tutorial http://www.phontron.com

[2] http://nlp.stanford.edu

dependencies as in figure 2.2.

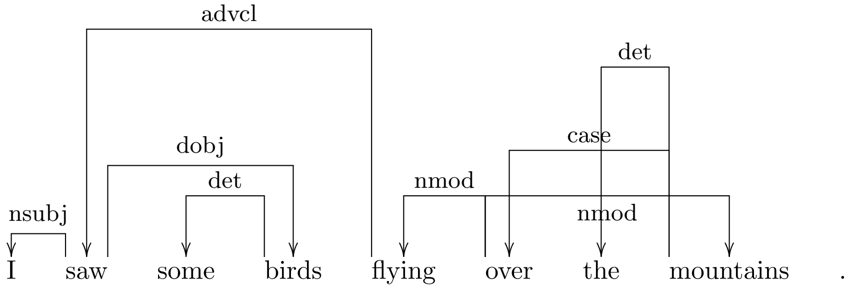


Figure 2.2: Typed dependency tree of the sample sentence: second point of view

The second main format of a parse tree-parse tree of structure-identifies forms of phrases in sentence and defines a hierarchy of root-branch-leaf structure for each phrase. In addition, the *Part-of-Speech (POS) tag* relevant with each word according to their role in the sentence is identified and visualized. In order to determine each tag used in this tree type, understanding *Penn part-of-speech tag set* is necessary [18]. Part-of-speech tagging is the process of identifying and tagging the grammatical and context-based role of each term in a sentence, which is also dependent on the adjacent terms [41]. A full-list of POS tags is available in the appendix A of this thesis. Considering an instance sentence with the class label "science":

"Researchers have successfully applied a theoretical medical treatment on brain cancer" (2.5)

Part-of-speech tagging is done using language model taggers who have been trained using millions of documents. Stanford POS tagger [42] for instance, returns a list of dual-tuples, consisting of each term plus the relative part-of-speech tag, as follows:

"[(Researchers, NNS),(have, VBP),(successfully, RB),(applied, VBN),(a, DT),(theoretical, JJ),(medical, JJ),(treatment, NN),(on, IN),(brain, NN),(cancer, NN)]" (2.6)

The value of POS tagging lies in the fact that each term is accompanied by an additional learning tag-or component-. These tags can potentially reorient the class attribute when added to each term. For instance, in *medical treatment* the machine initially learns that the type of *treatment* belongs to medicine and health sciences, when *medical* is added. Moreover, adding the POS tags of *JJ* and *NN* respectively, results in another sort of knowledge by defining a pattern. In this way, the machine can learn if the predecessor to the noun (NN) *treatment* is an adjective (JJ), this type of treatment is different than *requested treatment*, where the predecessor is not a direct adjective, but a form of verb, that can differentiate the treatment type. Utilizing POS tags can enhance the feature extraction, which is discussed further, specifically in chapter 6.

With the introduction of POS tags phenomena, we can continue with the second type of parse trees. Using sentence 2.4, a parse tree of structure-second type of parse trees-can be determined as follows in figure 2.3. In this figure two main compartments of the sentence which are the noun phrase (NP) *I* and the verb phrase (VP) *saw some birds flying over the mounts* are positioned at the root part of the parse tree. Breaking down the verb phrase will results in further grammatical compartments which are illustrated in figure 2.3.

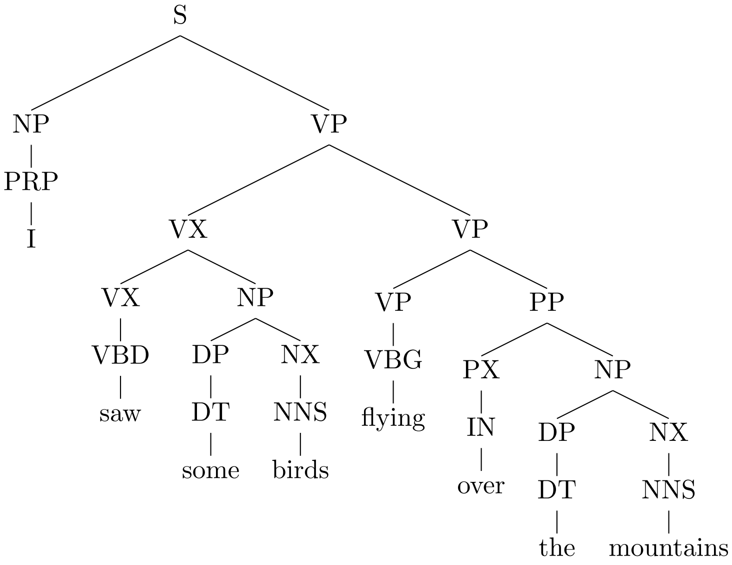


Figure 2.3: Parse tree of structure for the sentence in discussion with word roles

2.3.2 Punctuation marks removal and lower casing

The next step toward extracting informative features is the removal of punctuation marks for the ease of analysis. Additionally, normalizing text data is of great importance and as a result, a function transforms the whole document into lower case alphabetical letters. Hence, no conflicts will exist later on during the next analysis phase between *Archaeology* and *archaeology* for instance. Various methods in Python are able to remove punctuation marks. Some are built-in functions and are specific to domestic Python libraries including *String*. By using a simple script of:

```
''.join((x for x in string if x not in string.punctuation))    (2.7)
```

Python splits and joins later a clear string sequence without the punctuation marks list saved in the string.punctuation list. In the NLP specific library of NLTK [3], a certain method of $nltk \cdot wordpunct\_tokenize()$ is capable of removing punctuation marks. For transformation into lower case letters, the *.lower()* function handles strings in input and return the desired lower case output.

2.3.3 Stop-words

Removing stop-words of a language is a critical task in specific text mining tasks. Stop-words are the set of words that can be filtered as their significance is considerably low and the amount of uninformative search results they return is extremely high. These words are filtered during a text mining task [43]. A fixed list of stop-words for a language is normally not available as due to the context and means of analysis, excessive groups of words can be added to the stop-words list. For instance, consider the ministry of interior in a certain country is interested in understanding the relationship between the education level and marriage status of all people less than 40 years old and the level of salaries earned by these people. If data used in this study is assumed to be text data from ministry of welfare, words such as *health* and *gender* are of no importance as their significance is low in the context of the analysis. As a result, these words can be added to the normal list of English stop-words including *the*, *and*, *as* etc. For general purposes, Fox [44] has introduced a list of 421 most frequent and neutral stop-words in English which can be used in information retrieval applications.

In Python the list $stopwords \cdot words("english")$ has a predefined list of English stop-words. The user can specify the language of stop-words required inside parentheses. Hence, a simple script excludes the words in a word sequence whether they belong to the stop-words list.

[3] https://www.nltk.org/

2.3.4 Stemming

Stemming is a specific pre-processing task which is undertaken as a means of *dimension reduction* in feature extraction. Stemming is the task of reducing the number of derived forms of a root word and convert all into the original form. Although classification models based on stemming normally outperform others without stemming, it can be statistically insignificant compared to no stemming mode [45]. The original stemming algorithm by Porter [46] aimed to strip suffixes in English words automatically. Stemming is able to convert other forms of the word *Happy*, namely *Happily*, *Happier* and *Happiness* into the root word and by this act, remove or undermine the impact of these words on the analysis. Using *PorterStemmer* [47] method from package nltk.stem from NLTK in Python, users can stem a list of derived word forms into the original root form.

2.3.5 Noise removal

Numbers, hash tags, hyperlinks, special characters and HTML and XML tags are examples of noise in text data. These characters are normally removed using regular expressions. In most cases, one regular expression can be defined to remove all these characters in one step, to reduce the amount of memory required due to high load of text data and limited memory restriction of ad-hoc analysis.

In this respect, a typical text pre-processing pipeline can be shaped, although other techniques can be added or removed from the series of steps according to analysis requirements. Some examples include keeping numbers and special characters or handling uncovered word boundaries including apostrophe in manipulated words such as *wouldn't.* Figure 2.4 represents a hypothetical pre-processing pipeline achieved from self experience regarding the most used techniques.

Using the definition of a feature, text pre-processing to enhance data quality and feature preparation and finally state-of-the-art feature extraction methodologies, the output of a pre-processing pipeline is the input to the transformation function that converts the textual features to numerical features. At the most basic level, counting the number of optimized terms-using stemming-which is known as the *Bag of Words (BoW)* is the primary text to vector representation framework used in text analysis. Bag of words is discussed further in the next chapter. Yet, the final goal of similar methods is to transfer the textual features to a matrix-based representation space in a *Vector Space Model* which will be discussed in details in chapter 5.

In the next section, the role of word combinations and word markup tagging, to contribute in feature extraction is discussed.

2.4 N-grams and N-gram features

In many use cases, a set of documents which belong to a specific category, include token combinations that characterizes the class genre. Consider a group of docu-

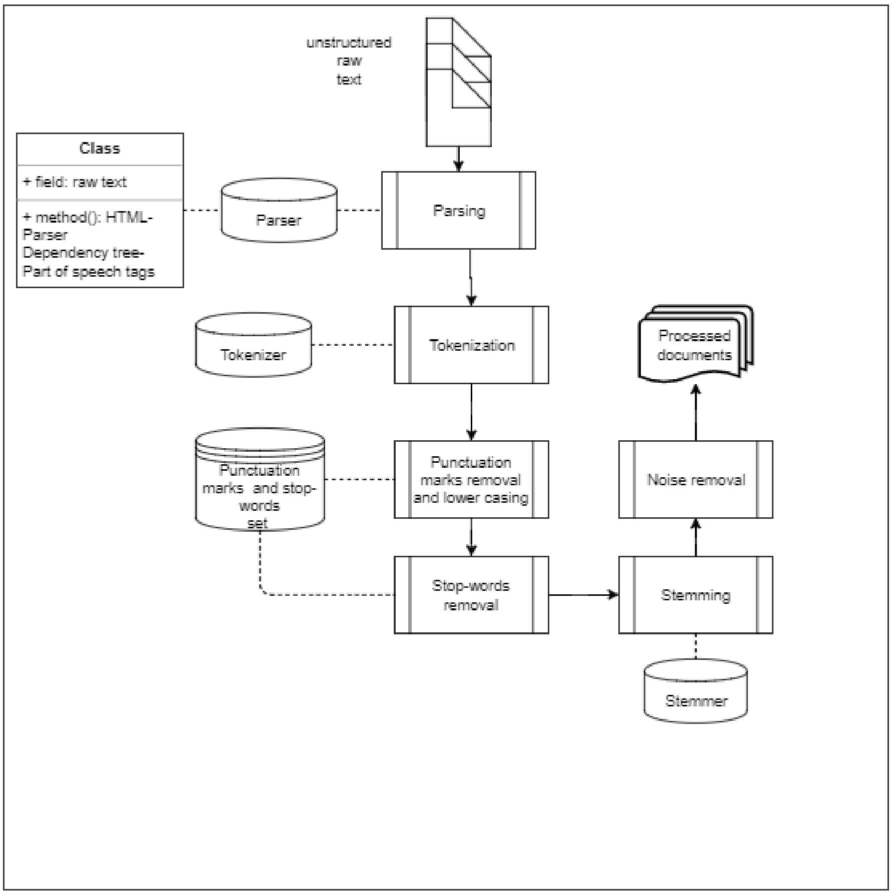


Figure 2.4: Conceptual model of a hypothetical pre-processing pipeline

ments which describe the developing technology in the passenger aircraft industry. Documents in this group include sentences which can be similar to the following examples:

"fuel efficiency and emission play two major roles in aircraft design"
(2.8)

and

"industry managers consider passenger comfort and noise reduction inside the cabin"
(2.9)

In these two sentences, the terms *efficiency*, *emission* and *design* individually correspond with a couple of subjects such as *performance*, *pollution* and *design*. However, considering the use of term combinations such as *fuel efficiency*, *major roles* and *aircraft design* provides the added value of concrete insight over the class, which in this case is more correlated with *industrial design* or *aircraft*. Standing on the higher levels of knowledge representation, are combinations of the length 3 or more such as *in aircraft design* or *emission play two major roles*. In the second sentence, the same methodology corresponds with a higher probability of class identification for term combinations such as *industry managers* and *noise reduction inside the cabin*, compared to *managers* and *noise*. The term combinations introduced in this regard are a type of *N-gram* features.

N-grams of text strings can characterize a document based on the frequency of terms. By definition, an N-gram is a continuous sequence of N entities from a language unit [48]. These units can be letters, words or even syllables. N-grams are important concepts in information theory and they provide knowledge in different applications [49, 50, 51]. Traditionally, for $N = 1, 2, 3$ the N-grams are respectively called unigram, bigram and trigram. For $N \geq 4$ the N-grams are named after N. In text classification, N-grams of words in documents are informative features. The fundamental informative aspect of using N-grams of letters, in English for example, is that with a higher probability, there is a chance that specific letters happen before or after some other specific letters. Language models based on probability distributions can determine specifications of text strings. An example of a document can visualize N-grams of words in different size:

$$\text{"\textit{inequality of salary levels among men and women}"} \tag{2.10}$$

By $N = 1$, unigrams include a sequence of one word at a time. As a result, 8 unigrams are formed as *"inequality, of, salary, levels, among, men, and, women"*. By $N = 2$, bigrams of words for the phrase include *"inequality of, of salary, salary levels, levels among, among men, men and, and women"* and finally for $N = 3$ we have the trigrams of words including *"inequality of salary, of salary levels, salary levels among, levels among men, among men and, men and women"*. N-grams of letters can be employed as informative features depending on the context of analysis as well and in many cases, these N-grams are preferred over the word N-grams. An important reason for this phenomenon is the manageable number of letter N-grams because of limited number of letters in each language. In this regard, generating letter N-gram frequency profiles and measuring the distance between equivalent N-gram profiles [48] can be determined to calculate the distance or in other words, the similarity of two documents, resulting in a text classification scheme. First, to define an N-gram frequency profile, we generate all N-grams of length $N = [1, 5]$. Next, a hashing table counts the number of occurrences for each N-gram. Finally a reversed order list of N-grams based on their counts forms the N-gram frequency profile. This means that the the frequency profile assigns a higher position to the lower occurring N-grams. Here, we can specify a restriction on the number of occurrences for N-grams to limit lower occurring combinations with low knowledge specificity. The acceptance range is the highest 300-500 occurring N-grams of subject. Since in this range, similar subjects include identical N-grams. For each reference document-reference profile-wich basically points toward a certain document class, a frequency profile is calculated in this way. In the next step, we rank the N-grams of the documents in comparison with a reference document and calculate a *sum of distance* measure for equivalent N-grams. These distance measures determine the *out-of-place* [48] score for each N-gram. Documents having a lower sum of distance measure with a certain reference document are assigned the same document class. A visualization of this case is represented in figure 2.5. Algorithm 1 summarizes the task as well.

For the example in figure 2.5, the out-of-place measure for *ESS* is 2, since it is situated in position 1 in reference document, while it is situated in place 3 in the test document. Positions are assigned starting from 0 and from left to right. For N-grams which are not available in the reference N-gram frequency profile, the maximum out-of-place measure is defined. Finally, the sum of out-of-place measure ranks documents based on their similarity with the reference document.

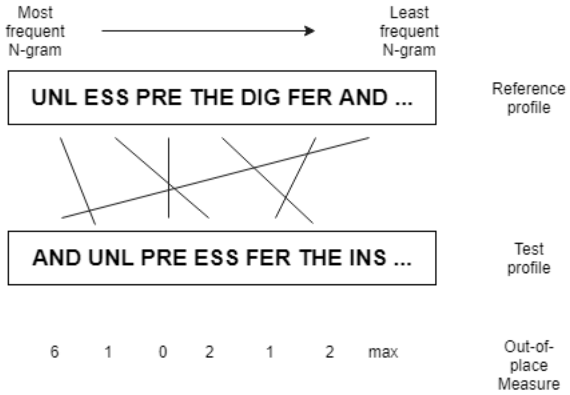


Figure 2.5: Out-of-place measure

2.5 Conclusion

Chapter 2 was devoted to the introduction of ground concepts of text classification and the relative terminology which is widely used in the next chapters of this

```
Data: Test document
Result: Degree of similarity between two documents using N-grams
initialization;
for a test document $d_t$ do
    generate N-grams for $N = [1, 5]$;
    for each set of N-grams do
        retrieve the class N-gram count;
        choose 300-500 highest occurring N-grams of the class;
        generate frequency profile for test document rank according to
          highest occurring N-grams;
    end
    for $t_d \in (t_1t_2t_3...t_n, t_2t_3t_4...t_n, ..., t_{n-3}t_{n-2}t_{n-1}t_n)$ do
        compare out-of-place measure with equivalent $t_d^{'}$ in reference
          document;
    end
    Calculate sum of out-of-measure for $d_t$;
end
```

Algorithm 1: N-gram frequency profile generation and out-of-place measure calculation

thesis. To perceive the methodology in addressing the second and third challenges and getting acquainted with the basic concepts of learning features, feature selection and extraction and the necessity of text entity transformation functions were elaborated in chapter 2. Different document parsing methodologies were introduced along with information on the effect of text pre-processing in returning more informative features and resulting in less required resources for the analysis task. In the next chapter, we provide the supervised machine learning fundamentals with respect to text classification and common feature selection approaches. Furthermore, justification of selecting a certain feature selection methodology in the next phases of this study is elaborated.

Chapter 3

Supervised learning feature management

Here in chapter 3 the fundamentals in feature selection for supervised text classification is introduced. We have represented learning feature selection methodologies in the context of text classification using previous insight. But first a quick look at trends in supervised ML applications which reveal patterns in text data is presented.

3.1 Knowledge extraction

Due to substantial use of text documents in business tasks, where documents maintain rules, knowledge, decision making approaches and instruction manuals, more and more demand has raised in knowledge extraction capabilities from unstructured text. In this regard, *Text Mining* is mainly focused on providing analysis infrastructure for data users to digest and understand semi-structured or unstructured data and to facilitate decision making based on the provided knowledge from these sources [52]. The problem of discovering patterns and meaningful information from text covers a wide scope of applications from the analysis subject point of view. *Unsupervised Machine Learning*, analysis at the *Named Entity Level*, *Topic Modeling* for text streams [53] and *Social Media Analysis* all specialize in certain fields of information extraction from text documents. In this chapter we are interested in the characteristics of text which provide potential in categorizing documents based on mutual features. With respect to text characteristics, currently two viewpoints exist in categorization approaches for text data. The first approach focuses on word-level representation of text documents while the second approach determines the semantics of words, entities and expressions. In this first viewpoint, normally a combination of words in a specific order determines and defines the spirit of the text. In this approach, relationships among words in text which are meaningful to human reader are not considered. Today, a huge portion of text classification research is built on such an approach [21]. On the contrary, semantics of entities in natural languages investigates the interrelations, dependencies and conjunctions among founding components. According to [54] in *Semantic Analysis* specialists move from superficial representations of language to a type of semantic metalanguage that can describe the inner layers of dependencies among language structure components.

The principles of data mining point to data stored in warehouses and databases, while current spread of knowledge in web necessitates text documents that rarely count as structured data. In other words, data mining techniques are able to analyze text data as long as such data is transformed into machine readable and structured formats. Text classification tends to employ data mining techniques bundled with transformation functions, in order to provide insight on similarities and develop categorization applications in text documents. During the last decade, applications such as Spam detection and search engine deployment through analysis of email content and search queries using classification of text strings spread in web technologies. In the first case, by inspecting a large number of email samples, specialists extract two lists of words, which are respectively used in spam and non-spam emails [55]. In the latter, a well-known approach generalizes keywords of queries from users and creates a *multi-leaf hierarchy* and assigns a number of nodes, in which all other incoming keywords are grouped and classified. Furthermore, documents will be classified according to placement in each certain node for keywords [56].

Recently, due to a growth in number of business intelligence fields in professions with large load of unstructured text data, more innovative applications are developed. In E-commerce, online trade and sales platforms categorize and present products and services according to search queries and further webpage visitations and followups. In other words, these platforms follow users journey from the first keywords typed for a specific product and then move towards other webpages until they find their preferred article or rather leave the platform [57]. The results of this journey help administrators to shorten the time for such a procedure, leading to a fast connection of demand with proper supply. Other applications include the integration of text classification for task assignment in Customer Relationship Management (CRM) platforms and marketing applications with understanding the trends in social media etc. Figure 3.1 presents a classification approach in a typical E-commerce platform.

As visualized in figure 3.1, queries issued by the human user can be in any structured or unstructured format. For instance, a certain user which is looking for a *milk frother* might not indeed know the correct word for the device that creates milk froth. As a result, the user might enter *milk twister* or simply *milk froth* as the search keywords. A smart E-commerce platform such as Amazon[1], processes the query and retrieves all similar queries which include *milk*, *froth* and all combinations made by these keywords. Then it compares the initial results

[1] https://www.amazon.com/

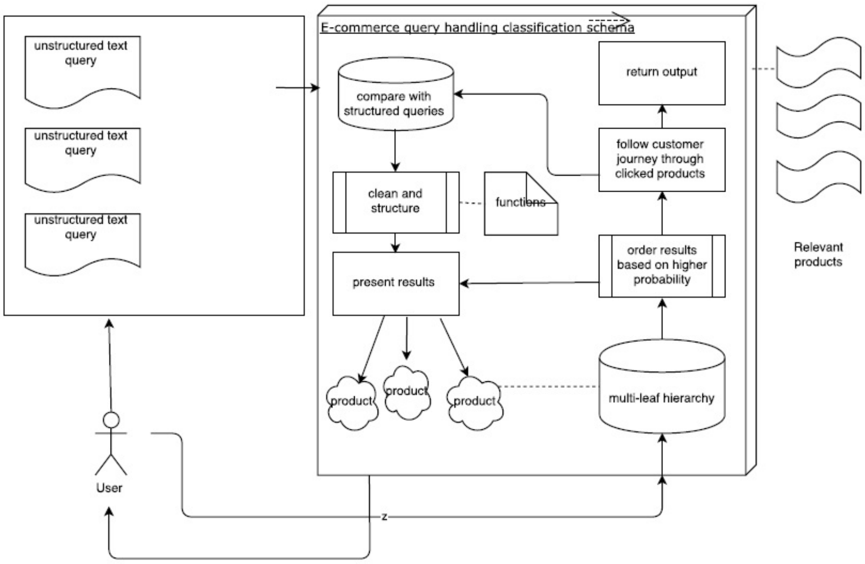


Figure 3.1: Classification in E-commerce

with the query and since such a query would probably return no direct product, it restructures the query by suggesting revised versions of the query. In addition, Amazon will return a series of products that are related to milk twisting, if the query is *milk twister*. Yet, as the E-commerce platform has previously seen similar queries that end up to a *milk frother*, it includes a couple of products related to this keyword. If the user is actually looking for something totally different than a *milk frother* such as a *milk warmer*, he/she would obviously search through the further pages of result products. In this way, the E-commerce platform finds out that the user is potentially looking for other products and using their multi-leaf hierarchy, jumps to products from the milk family, but with a further distance to specific *frother* and *twister* keywords. If on the further results presentation phase, the user clicks on one specific product returned by the multi-leaf hierarchy, the E-commerce platform would find out that a potential purchase case is created. As a result, it would continue showing similar products until the user selects a specific product. The complete set of interactions between the user and the E-commerce platform is known as the customer journey and the E-commerce platforms such as Amazon, record and analyze this journey to react in the best way in the case of similar future cases.

In all example applications mentioned above, the classes are defined and each class includes a specific set of features and characteristics which differentiates it from other classes. In the E-commerce platform for example, all products are internally classified based on many attributes such as customer type, use case, prices, etc. In addition to the previously classified set of documents in supervised text classification, the experts need to determine the classification features as stated in the previous chapter. In the next section we review the concept of feature selection in supervised machine learning and introduce a set of well known feature selection algorithms in this scope.

3.2 Supervised feature selection

The rudimentary distinctive characteristic of supervised ML text classification is the availability of class labels. Hence, there is a reference-or a set of references-of class attributes that enhances the classification model training. With a huge growth in text data size produced specifically in web content, and as far as documents include tens of thousands of terms, the number of features increases progressively. To overcome such a complication, in news classification for instance, the experts define a set of news topics such as *Politics*, *Science*, *Culture* and *Art*. Then, a set of labels for each topic would be determined. After the text cleansing and normalization, in case the feature selection is at term level, terms which potentially weigh higher than the others are analyzed and sorted out to be determined as the text classification features. A specific type of distance-or similarity-measure needs to be defined to rank the information level of features. In order to sort out these features, a couple of well known feature selection and extraction algorithms are introduced in the next sections.

3.2.1 Information gain

In case where the number of redundant features or terms which appear more than once are high, Information Gain (IG) is a proper feature selection method. IG is a measure of information in bits provided by each feature. A general IG function calculates the information the term t represents in prediction of the class c. This functionality originates from the fact that certain features are more capable of determining the class for a document. For instance consider two following documents:

"The support from the fans has considerably enhanced the motivation of the team" (3.1)

and

"The support from the managers has considerably enhanced the motivation of the team" (3.2)

Compared to 3.2, the document in 3.1 belongs to the *sports* category with a higher probability. In combination with the term *team*, the use of term *fans* corresponds with a sports team fans. Respectively, occurrence of term *managers* lowers this probability. Clearly, this is the effect of the term *fans* that results in selection of the class *sports* for 3.1, while these two documents only differ in one term. We can say, the information gain from *fans* has improved insight over class labeling for 3.1, compared to 3.2. Hence, a conventional information gain approach focuses only on the bits of information gained by each feature.

Moreover, a novel information gain approach combines two measures of information gained, first from each feature, and second the novelty of feature [58]. The second measure determines the difference between information gained by a certain feature and the features which are considered before. In this respect *Maximal Marginal Relevance* (MMR) proposed by Carbonell and Goldstein [59], states the fact that results returned for a user query are ranked based on their relevance to the query context initially. Further the measure of novelty can be added to the relevance ranking and result in a linear measure. The novelty of the result document corresponds with the least possible similarity to the previously retrieved documents. Maximizing the linear measure of relevance sorts all scores for each returned document.

Consider the document set $C = \{D_1, D_2, ...D_i, ...\}$, while Q is a certain query by a user, $R = IR(C, Q, \theta)$ is the sorted list of results as the output of the information retrieval function IR based on C, Q and threshold θ which is the minimum amount for which a document will be returned, S as the already selected subset of documents in R due to their relevance, $R \setminus S$ the difference between R and S, Sim_1 is a preferred similarity measure in document sorting-for instance term density, which is the density of occurrence of a specific term in the class documents to the total occurrence of the term in the whole set of documents [60]-and Sim_2 is the same as Sim_1 or a different measure of favorite-such as *cosine similarity*-, since the novelty measure is independent from the individual information gain measure. By the aforementioned definition, the following MMR formula is determined:

$$MMR = arg\,max_{D_i \in R \setminus S}[\lambda(Sim_1(D_i, Q) - (1-\lambda)max_{D_j \in S}Sim_2(D_i, D_j))] \tag{3.3}$$

The parameter λ is set by the user between 0 and 1. As far as λ tends to reach 0, novelty of the document compared to the already selected documents declines, as $1 - \lambda$ grows to reach the maximum. Increasing the amount of λ provides the results which diversify as λ grows, resulting in a more specific result document.

Accordingly, a feature selection method definition based on information gain from each feature can be framed [58]. Having C as class categories, R as the ranked list of documents returned by the IR system at hand, S and $R \setminus S$ are respectively the group of already selected and not already selected features, IG as information gain and finally $IGpair$ as information gain from determining feature pairs, $MMR - FS$ can be defined as follows. FS corresponds with the feature selection task.

$$MMR{-}FS = arg\,max_{D_i \in R \setminus S}[\lambda \cdot IG(w_i;\ C) - (1-\lambda) max_{w_j \in S}\, IGpair(w_i;\ w_j) \mid C] \tag{3.4}$$

With simpler words, 3.4 states that the measure of competency for a text classification feature is maximizing the amount of information gained from the feature itself, minus the information gain of novelty of the feature. Hence, a feature which has distinctive information over the class but has occurred in the previously selected document in the class, is relatively less important compared to an informative feature with less previous occurrences. The general reason is the tendency to improvement, which requires identification of new distinctive text features.

In the above definition, *IG* and *IGpair* are broken into the constructive elements, as follows:

$$\begin{aligned} IG(w_i;\ C) = &-\sum_k p(C_k) \log p(C_k) \ + \ p(w_i) \sum_k p(C_k \mid w_i) \log p(C_k \mid w_i) \\ &+ p(\bar{w}_i) \sum_k p(C_k \mid \bar{w}_i) \log p(C_k \mid \bar{w}_i) \end{aligned} \tag{3.5}$$

$$\begin{aligned} IGpair(w_i;\ w_j \mid C) = &-\sum_k p(C_k) \log p(C_k) \ + \ p(w_i, j) \sum_k p(C_k \mid w_i, j) \log p(C_k \mid w_i, j) \\ &+ p(\bar{w}_{i,j}) \sum_k p(C_k \mid \bar{w}_{i,j}) \log p(C_k \mid \bar{w}_{i,j}) \end{aligned} \tag{3.6}$$

In definitions 3.5 and 3.6, $p(w_i)$ and $p(\bar{w}_i)$ respectively point to probability of occurrence and absence of word-or feature- w_i. $p(C_k)$ is the probability of class k assignment and $p(C_k \mid w_i)$ is the probability of class k assignment given word w_i occurs. Also, $p(C_k \mid w_{i,j})$ corresponds to probability of class k assignment given the co-occurrence for words w_i and w_j, while $p(C_k \mid \bar{w}_{i,j})$ defines the opposite case. In both definitions, all forms of

$$-\sum_n p(X_n) \log p(X_n) \tag{3.7}$$

correspond with the *Information Entropy* [61] for each term. Information Entropy is measured in bits and is the logarithm of the probability distribution of an observed event. For instance, the Information Entropy of the condition of a light bulb is 1 bit, since it can only be in one of the two conditions of *On* and *Off*. For more than one observations of each event which takes the amount of n outcomes and n is a power of 2, $\log_2(n)$ expresses the bits of information for this series of events. Since the probability of occurrence of some events might be higher than the other outcome condition, the less probable event is a more informative feature, and as a result the Information Entropy of all data is $\log_2(n)$ at maximum. In this respect, $IGpair$ is defined as the information entropy of class C_k plus the product of information entropy of conditional occurrence of C_k given the simultaneous occurrence of term w_i and w_j and probability of occurrence of both terms plus the information entropy of the opposite case, which means these two specific terms do not occur at once.

As an example to have a more clear view on MMR feature selection, we can presume a collection of documents including 5 documents $\{D_1, D_2, ..., D_5\}$. MMR is iterative. Hence, all documents are retrieved by the IR system and initially listed in the R/S. As a result, S is empty initially. During the first iteration, document D_1, assumed to be the highest ranked document, is assigned to D_i in formula 3.3. Assuming λ to be any amount between 0 and 1, we calculate the similarity of a new query Q to the D_1-using any similarity measure like term density or cosine similarity-and since there is no other document in S already, the second part of the formula which represents the similarity of document D_1 and other document will be zero. Note the effect of λ in giving weight to similarity of a document and the query along with the offsetting effect on similarity of the documents which are already selected. In the next iteration, D_i will be D_2 if we assume it to be the second on the ranking. This time, the weighted similarity of D_2 and D_1 will be subtracted from the weighted similarity of D_2 and Q. In the next iterations, the maximum similarity measure of the new documents D_3, $D4$ and D_5 is calculated. This means that after first two documents are put in S, for D_3 we need to calculate if the similarity between D_3 and D_2 is bigger that similarity of D_3 and D_1. This case holds for the remaining documents as well. In this way MMR finds the degree of significance of a document and subtracts the value of being already known to the system from it. The use of λ adjusts the the trade-off between accuracy of similarity between the query Q which needs to find similar documents and the level of diversity of the resulting documents returned. The more diverse the results can possibly be, along with a considerable level of accuracy, the results are better in quality.

Information gain by each feature is a relative measure of amount for λ. Hence, for all the features present in a document, a ranked score list can be framed using the information gain, employing the co-occurrence of words capability in MMR-FS. As a result, further feature selection based on the score that each feature is presenting can be undertaken. In such cases, an assumed threshold for the number of features with the highest information gain can be considered. As an example, the first 1000 features with the highest score can be chosen.

3.2.2 χ^2 measure

Chi-square is a statistic measure which can define an attribute of independence of the class to a feature and hence, be used in feature selection. Frequency of features plays a major role in running the chi-square test. The reason lies in the fact that Chi-square feature selection considers the term i and class j and further creates the contingency table for each pair of $t_{i,j}$, where the amount for each entity will be the sum of frequencies for item i across class j. Using the number of times each term happens in the document, the sum will be calculated as the total number of times each term happens in class. Using chi-square test, the independence of the term and the class will be calculated and finally a ranked list is created. This means if the dependency between a feature or a term is relatively high, similar terms can be related to occurrence of the class. In other words, as chi-square test assumes the independence of the feature and the class-null hypothesis-, if the test reaches a significant amount for χ^2, then the null hypothesis is rejected and the feature depends on the class. Similar features can be selected for training the model. Equation 3.8 shows the chi-square calculation:

$$\chi^2 = \sum \frac{\left(o_i - e_i\right)^2}{e_i} \tag{3.8}$$

Where o_i is the observed value and e_i is the value expected for the entity - the feature - in feature extraction. While this equation might not be informative enough for the feature selection problem due to limitations in the observed and expected values for each feature, a modified equation is introduced according to [62]:

$$\chi^2(t_i, c_j) = \frac{\sum [P(t_i,c_j)P(\bar{t_i},\bar{c_j}) - P(t_i,\bar{c_j})P(\bar{t_i},c_j)]^2}{P(t_i)P(\bar{t_i})P(c_j)} \tag{3.9}$$

The above equation suggests that the independence between term t_i and class c_j is a function of the second power of sum of observed features which is the occurrence of the term t_i and class c_j and the opposite case where not of the two constraints are true minus the expected value which includes occurrence of term t_i but not the class c_j and the opposite case, divided by probability of occurrence of the term, probability of absence of the term and the probability of the class.

As an example consider a document class c to be *disappointing* where term X to be *deficit* occurs as a feature, chi-square assumes the null hypothesis that term X is independent of class c, or in other words, not relevant to the category. Clearly, the observed value of the simultaneous occurrence of the term and the class and the opposite case is relatively high, since by occurrence of the term *deficit*, occurrence of the class *disappointing* is probably high. The occurrence of the opposite case is probably high as well. However, the expected value on the second part is relatively smaller than the first part, as occurrence of the term *deficit* and a class such as *satisfying* is less probable, in addition to the opposite case, such as absence of the term and occurrence of the class *disappointing*. Now, by calculating a significant amount for χ^2, the null hypothesis is rejected, which means term X is relevant to the category c. Also, the amount of χ^2 at hand for each feature in test document can be sorted to rank the most relevant features to the category. Having a two class feature selection problem, chi-square formula in 3.5 can be broken down as follows. Let x and y be the classes in question, where the number of times a certain feature of F appears in class x is X and the number of times feature F appears in y is Y. Respectively, the number of data instances in x that does not contain F is Z and the number of instances in class y that does not include F is defined to be W. Hence, for each instance group X, Y, Z and W, the expected value can be calculated based on 3.8.

$$\chi^2 = \frac{(X - E_X)^2}{E_X} + \frac{(Y - E_Y)^2}{E_Y} + \frac{(Z - E_Z)^2}{E_Z} + \frac{(W - E_W)^2}{E_W} \tag{3.10}$$

Considering the total number of instances to be N, while the total number of instances containing F in both x and y correspond with $X + Z$, the number of instances lacking feature F will be defined as $N-(X+Y)$. Using similar definition, with the total number of instances in class x to be $X + Z$, the total number of instances in class y would be $Y + W = N - (X + Z)$. Modification of equation 3.8 using above definitions, is translated to finding out chi-square measure for the data sample, given the total number of instances in the data set, the total number of samples for a class that contain the feature F and the total number of instances from both class containing feature F.

3.2.3 Term frequency-Inverse document frequency

Although determination of term weighting as a measure of class definition was introduced by Luhn [63] in 1957, the concept of term frequency (TF) has been widely used in text mining applications and a big number of suggestion systems and search engines rely on modeling capabilities of a modified version of term frequency [64]. Term frequency is built on a basic concept that states that a document inherits the subject of reoccurring terms. In other words, terms that occur frequently have a higher probability of defining the class of the document. Hence, the frequency of terms-or the sum of times a term appears-can be nominated as a feature selection methodology. Moreover, terms in a document will be weighted simply based on the number of times they appear in the document. Consider the following four documents in table 3.1.

Table 3.1: Example documents

Document number	Document
1	Chicago is much colder than New Mexico
2	Chicago is smaller than California
3	New Mexico is not warmer than Tijuana
4	Tijuana is bigger than California

A binary vector representation of the four documents only determines the number of times each term appear in each document. The order of the terms is neglected in this view point. As a result two documents of *"Chicago is bigger than New Mexico"* and *"New Mexico is bigger than Chicago"* are detected identical. Table 3.2 visualizes a binary term representation for the four documents above.

Document	Chicago	is	New Mexico	bigger	than	colder	Tijuana	California	warmer	smaller	not	much
1	1	1	0	0	1	1	0	0	0	0	0	1
2	1	1	0	0	1	0	0	1	0	0	0	0
3	0	1	1	0	1	0	1	0	1	0	1	0
4	0	1	0	1	1	0	1	1	0	0	0	0

Table 3.2: Binary term representation for the four documents

Such a representation evolves further into the *Bag of Words* language modeling [65]. Bag of words (BoW) assumes a bag containing all terms present in a set of documents. Respectively, each document will correspond with a vector, depending on occurrence or absence of term coming out of the bag in that document. As a result, the order of occurrence for terms in different documents has no value for the means of analysis. Hence, specific characteristics that can classify a document exist in the occurrence of terms.

BoW provides a binary vector for each document which can be further emerged with a weighting framework based on the number of occurrences of terms. Such an ontology gives more weight to frequent terms which transcend document to document. In this regard, BoW suffers two critical weak spots while dealing with a substantial set of documents. First, with a big number of terms, there will be a high number of vectors with too many zeros, due to absence of effective and rare terms which actually correspond to the subject of the document. Second, each language includes a number of frequent terms which transfer no information value. In English for example, terms like *"the"*, *"a"*, *"for"*, *"so"* and many others exist with minimum knowledge value to classification problem. We introduced these terms as *Stopwords* in the previous chapter. As expressed before, the focus is generally on removing these terms in a text mining application. Due to BoW structure, higher occurrence recorded for such terms will add weights to them, resulting in models trained based on low value terms and deficit categorization capabilities.

It is necessary to mention that the relevance of subject to the term does not increase by the same proportion. For instance, a document with four times occurrences of the word *ball* is more probable to be about *sports.* But this probability is not four times higher than a document with one occurrence of *ball.* Also consider two documents, where the first document contains 100,000 occurrences for the term *ball* and the second contains 200,000 of the same word. Here the relevance of the first document to sports is still very high. That is why it is not possible to say that relevance of the second document to sports is twice as the first document. For a document d having a specific term t, term frequency of the term is defined as $tf_{t,d}$.

As a result, a modified form of term frequency weighting scale has been introduced based on the *log frequency* of terms instead of raw term frequency. Definition of the log frequency used in weighting scale follows:

$$w_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d} & \text{if} \quad tf_{t,d} > 0 \\ 0 & \text{if} \quad otherwise \end{cases} \qquad (3.11)$$

Term frequency-inverse document frequency (Tf-Idf) aims to offset the effect of frequent terms while magnifying the weight of very few rare terms with categorization characteristics. This means for example in the case of weather forecast, while terms such as *warm* and *cool* should be assigned lower weights as frequent terms, terms such as *storm* and *hurricane* will be assigned a high weight as they have a higher relevance to weather forecast than the two first terms and they relatively count as rare terms. In Tf-Idf approach, the inverse document frequency is an inverse function of the number of times a term occurs across the whole set of documents [63] and is calculated with dividing N as the number of all documents, by the number of documents in which term t appears.

$$Idf(t, D) = \log \frac{N}{\parallel \{d \in D : t \in d\} \parallel} \qquad (3.12)$$

and since Tf-Idf is the product of $tf_{t,d}$ and $idf_{t,d}$:

$$Tf - Idf_{t,d} = (1 + \log_{10} tf_{t,d}) \times \log \frac{N}{\| \{d \in D : t \in d\} \|} \tag{3.13}$$

In order to have a more clear view over Tf-Idf approach consider a document with 1000 terms where the word *ball* occurs 20 times. The human reader interprets that these documents might be related to sports as *ball* is not a frequent term in English and it has occurred a considerable number of times. As a result term frequency of the term *ball* will be 0.02. Respectively, if in a set of 100,000 of documents, the term *ball* occurs in 1,000 documents, Idf will be calculated as:

$$Idf(t, D) = \log \frac{100000}{1000} = 2 \tag{3.14}$$

Consequently, $Tf - Idf$ will be $2 \times 0.02 = 0.04$. The aforementioned equation presents how the effect of frequent words is biased with their low weight, as in the case of a certain word *this* and assuming 100,000 of documents where *this* occurs for fifty thousand times, the product of $\log \frac{100000}{50000} = 0.3$, which is lower than the amount of Idf in the previous case. Clearly lower weight will be granted for Tf-Idf for this term, assuming the same conditions like the first case, with 20 occurrences of the word *this* in one document with 1000 terms. Eventually, weights for each term-feature-can be ranked and a specific number of features with a Tf-Idf score of higher than threshold will be selected for model training. Further in the next chapters, we have introduced how and if Tf-Idf fits in a certain case study.

With introduction of the ground concepts in building Tf-Idf, in the last section of this chapter a very widely used term mapping technology in semi-supervised text analysis approaches is introduced.

3.3 WordNet technology

In this last section, we briefly present a comprehensive lexical technology which is widely used in determination of word dependencies. *WordNet* is a network of words including more than 117,000 entries, representing nouns, verbs, adjectives and adverbs and their synonyms, antonyms and hypernyms along a cognitive map of each attribute and word senses. These entities are called *synsets* of a word [66]. WordNet covers the relationship between word synonymy-such as *speech* and *dialogue*-, antonymy-such as *anxious* and *bold*-, and hyponymy (hypernymy)-such as *apple* and *tree*-. The significance of a network dictionary technology such as WordNet lies in the fact that using one refereed lexicon, many applications in text mining such as text classification and sentiment analysis can broaden their word sense knowledge. Not only by considering a set of words as analysis infrastructure a user can define all relevant words and categorize dependent statements according to the relationship, but also a similarity measure of concepts is at hand using WordNet [67]. Considering a relationship between a *lamp* and the *light*, WordNet is able to detect the relationship between an *engine* and *power* as power is deeply connected with attributes of an engine such as move, speed and acceleration.

The latest version of WordNet 3.0, provides graphical user interface and plugins for major software development environments. A user is able to submit a query for a word to see a network of hyponyms. WordNet supports both *direct hyponyms* and *full domain hyponyms* which respectively represent direct combinations of a word such as social psychology for psychology and psychophysics which is a branch term of experimental psychology, the main branch of psychology.

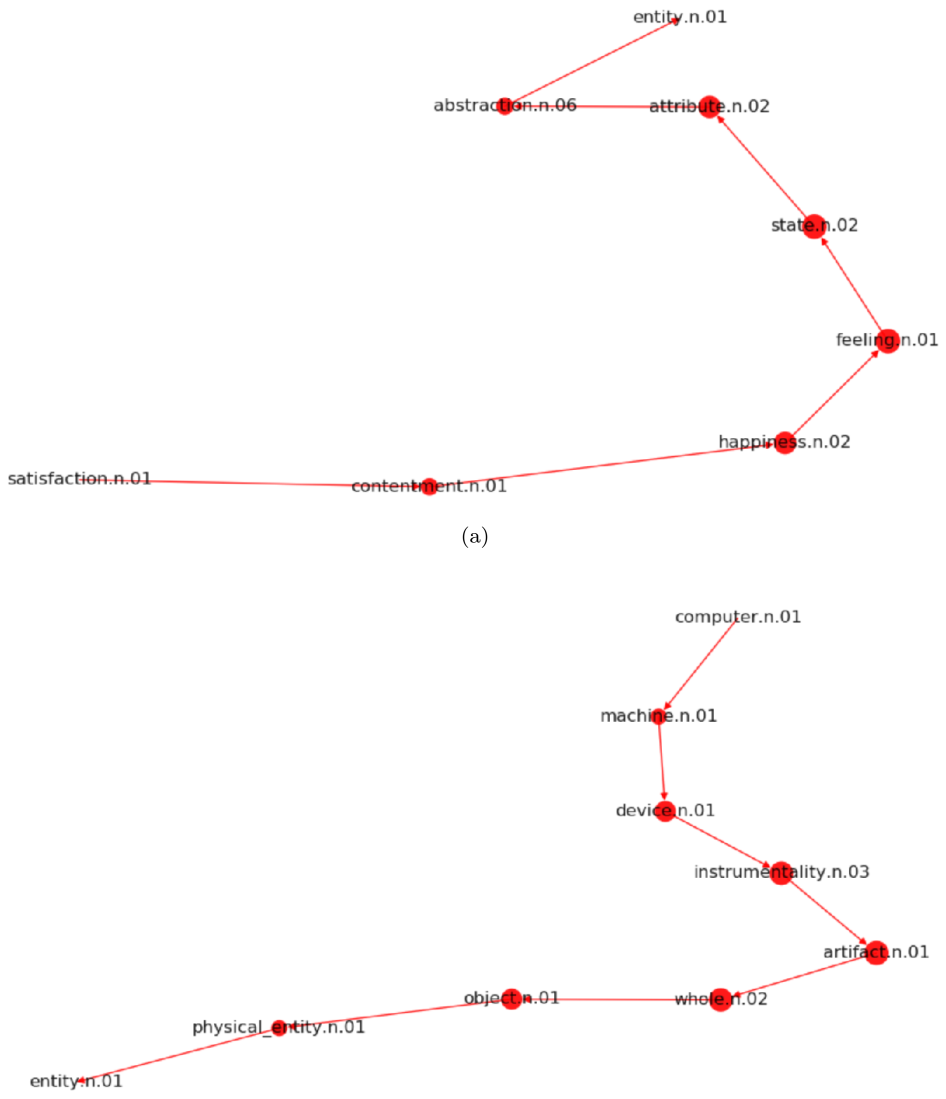
As an example consider the hyponym network of two random words *satisfaction* and *computer* generated by embedded WordNet plugin for Python and presented in figure 3.2. The relationship for each word visualized in these graphs presents the superordinate conjunctions to the leaf word-i.e. in the relationship between the *chair* and *furniture*, furniture is the superordinate term-. Each synset is connected to its superordinate via a pointing arrow. The structure of each synset in the figure is composed of the form *lemma.pos.number*. *lemma* is the leaf synset of a certain superordinate term. *pos* is the pos tag or the grammatical position of the lemma, which is *n* in the figures, corresponding to unknown or null. *number* defines if each leaf is at the same level with their superordinate leaf term. For instance in figure 3.2(a), *feeling* and *state* are all leafs which are situated on the same level. Hence, they receive ordinal values of 01 and 02. If there was necessary to have another term such as attitude in the map, it would probably receive the number 03. The ending point of arrow represents the superordinate. These figures are generated in a random way each time the hyponym network is asked again and specific movement directions of the graph do not imply any meaning.

In the experiment represented in chapter 4, we tried to employ WordNet as a lookup function to extract all synonyms and hyponyms of a word. However, due to applicability restrictions to the task and the dataset, the use of WordNet to generalize common root words was not successful. Further specifications and details are presented in chapter 4.

Chapter 3 focused on the implications of supervised feature selection methodologies. In the next chapter we have presented a case study of implementation of supervised text classification for a parcel delivery company to test the methodologies presented in this chapter and chapter 2.

3.4 Conclusion

Enabled with information over the class characteristics and pre-classified set of documents, supervised machine learning is able to deal with corpus of text documents with any number of features. In this respect, we reviewed the concept of features in text analysis and discussed a couple of highly in-use feature selection methodologies. Presentation of each methodology clarifies text characteristics and develops infrastructure for hybrid feature selection methods, that can overcome specific type of text documents, as we have introduced and analyzed in chapter 4. Following a comparison of the three introduced feature selection methodologies, advantages of the Tf-Idf as the chosen methodology over the others were expressed. Feature selection methods facilitate the extraction of informative characteristics of text, in order to train classification models using supervised machine learning approach. A case study employing text pre-processing and complexity breakdown for dual language text documents is presented in chapter 4 based on the concepts elaborated in chapter 3.



(a)

(b)

Figure 3.2: Hyponym WordNet of two random words computer(a) and satisfaction(b)

Chapter 4

Unraveling Bi-Lingual Multi-Feature Based Text Classification: A case study

Chapter 4 of this thesis, implements theories and frameworks of supervised text classification on a real business intelligence case. The goal is to develop a classification framework for a database of parcels based on their description attribute, which are text documents. This chapter has been published as an article in the refereed scientific journal Information Theories and Applications [28].

4.1 Introduction

Research on applications related to text pattern recognition in business operations has lead to solutions deepened in sentiment analysis applications in marketing, social media mining for product and service penetration, fraud detection in financial statements, customer service and many more. Unique potentials of pattern discovery in documents, from structured to totally unstructured, empowers business administrators to extract rules even from informal and manipulated sentences. However, applications in this context are sensitive to the text structure and vary on the use and the expected results. An attractive case study for companies producing and transferring commercial goods is to employ a classification system for the items they produce, send or receive. Such a system could be highly valuable to inbound and outbound logistic services as well [68]. The combination of text mining and supervised machine learning is known as corpus-based document classification. Furthermore, in this context, documents are being annotated manually in order to create corpora, that act as a relational database of documents, categorized and supervised to be used later on, to assess and analyze consecutive document datasets.

Commercial commodity categorization using text mining implements a classification framework using pre-labeled items to train a classifier [69]. Moreover, the ontology behind supervised text classification highlights the need for enough data to train the classifier. In other words, problem solving cases dealing with commodity classification focuses on using commodity attributes to group similar items.

The phenomenon of Supervised Machine Learning has been introduced to business decision making in several domains in order to simplify tremendous complex datasets which represent few or no guidelines in data interpretation. On the other hand, more and more organizations are interested to apply such frameworks to documents in which lies patterns that can be modified towards growth and clarification. Text mining, which intensifies the hidden structure in documents, is recently well integrated in classification functions.

Representation of the case study in this research focuses on the application of current text classification techniques on a common data classification problem in business context. The problem we are going to address in this chapter comes from an international parcel shipping company in Middle East. The company has provided a dataset of delivered parcels with tens of thousands of records. Each record in the first dataset has a distinctive case id with another 55 attributes. The complete dataset contains more than a million records. Additionally, there is a list of 70 commodity classes, exclusively generated by the company. Part of the original dataset of delivered parcels is already classified using the list of commodity classes. The company is interested in categorization of further products which lack the commodity class. This means that all records from the delivered parcels dataset can be assigned one of the classes from the 70 classes. The 70 groups include classes such as *household appliance*, *cosmetics* and *computers.* A partial list of 70 categories can be found in figure 4.3. The complete list though is classified data by the company and cannot be published. The parcel delivery records dataset include attributes which are almost exclusively about the route, weight, size and sender and receiver address.

Parcel delivery companies are not normally able to open packages to get insights into the contents. Yet, one of the data attributes which can indicate the required class is the attribute of *commodity description.* Commodity descriptions are provided by the merchants that act as package senders. This type of information is represented using various languages. Distinctive terms compose these descriptions which include product technical descriptors, use cases, gender specifications, material, etc. The company requires a classification scheme to group all relevant items to one of the 70 groups provided. Among these groups, 69 categories are specific categories such as we mentioned above. The last group is dedicated to all items which cannot be associated with any of the remaining groups. This last category is called *others.* As stated before, in text analysis domain, except for the commodity description which potentially includes terms which belong to one of 70 predefined groups, none of the data attributes provide any insight regarding the classes. Except for data attributes which contain abbreviations regarding product type or product group such as *DOM* or *CODS*, which has specific meaning for the parcel delivery company, the rest of the data attributes are numerical. As a result, matching any of the commodity classes to these data attributes is not possible. Refer to figure 4.1 for a better image.

Since the parcel shipping and delivery company is situated on a strong position among other rivals, the knowledge extracted from such a dataset would enable them to benefit from a number of advantages. By understanding how many of each category they have delivered, they can be identified as a trusted and efficient delivery partner of specific items which are valued to be received well protected and in a reasonable time at destination. Examples include jewelry and electronic devices which are of high value for the sender and receiver. Additionally, they can provide the insight over destination and customer segments in each region which have acquired certain commodities to interested merchants. The merchants can then focus their marketing efforts on these segments and regions to enhance their sales. For instance if the parcel delivery indicates that cosmetics are mostly sent to a town in north Emirates, the merchants would identify a potential buying segment which would be targeted for marketing campaigns.

We present a methodology to address the question of commodity classification by dealing with parcel delivery records with a commodity description attribute in two languages. As each business around the world manages to handle a variety of databases with terms in two or more languages, a proactive text classification approach with respect to text parsing for a proper machine input should be taken into consideration. In other words, the experts should resolve the problem of datasets with multilingual terms. We are proposing a three step framework to analyze and categorize products in 70 commodity groups provided by the parcel delivery company:

1. Data cleansing, pre-processing and translation.
2. Feature extraction using a term-weight based ontology.
3. Classification model proposition and testing the results.

To unravel the dual-or multi-lingual text data dilemma, one approach might be to separate records from each language and proceed with the analysis with more than one language model and to aggregate the results further. Yet, NLP capabilities with respect to tokenization, parsing, part-of-speech tagging and the availability of semi-unsupervised text mining facilities such as a *dictionary* in different languages varies and is not the same. Furthermore, NLP environments are not defined based on standard criteria nor by a single developer and hence, the results might be unexpected. Additionally, some records might include terms from two or more languages which makes it impossible to analyze each set of language specific terms separately. The added value of our approach is to translate the records into a unified language model and to reduce the number of features for a higher precision in category prediction with a high degree of applicability in commercial data mining.

As a case study of potentials of supervised machine learning, the main objective of this study is to classify records-and commodities as a result-in big datasets of delivered commodities including dual languages of English and Arabic and to normalize the data records using categorization into a set of pre-defined classes in company's interest. Although employing a pre-defined dictionary of terms seems reasonable to undertake in order to reduce the number of terms-for instance aggregating all sorts of words such as laptop, PDA, notebook, etc. to computers-, we faced limitations on implementation of such an approach which would be discussed further.

The rest of this chapter is organized as follows. Section 4.2 will go through recent study and evidence applying text mining techniques in text classification. Section 4.3 focuses on the proposed framework considering the structure of data in this study as the input. In section 4.4, model selection is discussed in detail. The results of the proposed framework have been displayed in section 4.5. Finally, section 4.6 outlines the conclusions.

4.2 Related work

Commodity classification has been traditionally explored from the quantitative attribute analysis point of view which reports on the numerical data attributes such as weight and size. Examples include prediction of necessary means of transport or the insurance cost for incoming commodity in case of transporting the commodity, based on the previous records. Distinctive criteria in this regard include the value of the commodity, the size, the weight, the destination, etc. However, in this study, the focus point is classifying commodities based on textual attributes and determine commodities correspondence with one of a set of predefined classes. Underlying qualitative aspects such as use case, material, the gender based customer segment, etc. characterize the commodity description in this regard. Features extracted from commodity description attribute refine the classification task in this study.

Very few studies have focused on industry or service-based text data classification to provide insight or ideas in further research such as commodity classification. In [70], the authors have employed association rule mining on Post Project Reviews to enhance industrial knowledge management. Ghani et al [71], have represented the idea of product identification based on a series of attribute-value pairs. Such a methodology allows feature vector extraction that supports the term frequency framework in text classification.

Popescu and Etzioni [72] have built the model called OPINE based on the review scores and features extracted from customer reviews. A classification framework based on these features can be constructed. In all of these studies, selecting a relative learning feature has proved to be of significant importance.

Mapping terms to reduce a great number of features like would as well enhance the feature selection procedure, with a similar impact to stemming. Recently the application of *WordNet* in lexicon-based text classification has created opportunities in semantic mappings in product classification [73]. Unlike lexicon-based text mining approaches which employ dictionaries of word trees, with broad synsets-which are synonyms for English words-, a total supervised classification approach employs the insight gathered from previously classified instances. In this study we will focus on the supervised approach and elaborate on the benefits of this approach compared to the unsupervised dictionary-based approach.

4.3 Feature selection methodology

In the experiments presented in this chapter and chapter 6, documents to be analyzed are of two different types. In this chapter, we analyze documents which are relatively short, including less than 20 terms per document. The terms express details over a set of attributes such as size, weight, material and capacity of a set of products. Additionally, with availability of predefined classes and training data, Tf-Idf can compute the similarity of a document with a class reference document with relatively lower efforts compared to Information Gain and χ^2. In the second experiment presented in chapter 6, the documents are larger in size compared to the first experiment, since we are dealing with sentences rather than descriptive expressions. Yet, both types of documents are similar in terms of reduced number of uninformative features using text cleansing, while due to the diversity of the set of terms used in both document sets, the feature set matrix is still of high dimensions.

Information Gain and χ^2 would struggle with occurrences of high number of features. Specifically in case of Information Gain, there is a need for insight on conditional occurrence or absence of certain features, while such an insight is currently not available. In our current dataset, the terms occur independently from each other. IG counts on occurrence or absence of each term $t_i - 1$ in amount of information that respective terms t_i and $t_i + 1$ gain sequentially, in predicting the class category. As a result, IG would act poorly in this experiment as there is no information on serial occurrence of certain terms. For the case of chi-square feature selection, one needs to calculate the conditional probability of absence or presence of a term given a certain class in addition to conditional probability of occurrence or absence of the term, given the absence of the certain class. χ^2 as well rules for only a predefined set of features from the whole dataset in addition to only one degree of freedom. According to Schütze et al. [74], with substantial number of features, the number of times a statistical test should be performed multiple times. In this way, to reject an assumed number of 10,000 null hypothesis-to prove the dependency of a term to class-, with an assumed error rate of 5%, an average number of 500 tests will be wrong, which is a considerable number of trials. Hence, the number of used features in χ^2 should be limited, which limits the task of analysis in both experiments, with a big number of documents, terms and respective features.

Taking into account the limitations that IG and χ^2 impose, we have decided to focus on Tf-Idf for the next two experiments. Although Tf-Idf misses the insight over the semantics of the features in text classification-which is the interrelationships of features with respect to subject and the named entity-, it can capture the information from any number of documents and features and it has proved to be efficient in supervised text classification [75].

4.4 Research methodology

We propose a supervised classification methodology for commodity classification based on commodity descriptions. Our input data is composed of thousands of rows of documents-commodity descriptions-in English, Arabic or a combination of both. As the first step, we convert every document into English to normalize the data input. Next, the pre-processing pipeline is applied on the training data to clean unwanted information and prepare the text data to be transformed into numerical features. Each commodity description is composed of a limited number of normalized terms after pre-processing, which we would employ and transform into the term matrix. The document-term matrix is then created using Tf-Idf as stated before, to signify the rare but important terms in all documents in the dataset.

With a clear view over selecting Tf-Idf to compose the term-document matrix and to select/filter the most informative features, we divided the dataset presented by the parcel delivery company into training and test sets. 80% of the whole data is assigned to the training set, while the remaining 20% is assigned to the test dataset. Different classification algorithms are then trained using the term matrix on training dataset. The trained models are then tested using the test dataset and the precision of the models, their recall and F-score is calculated. Hence, a classification schema is formed which is able to predict categories of future parcel delivery commodities, presuming the availability of pre-classified data. The major advantage of this study is the ability of the framework to handle multi-language records with a high number of features with an application in commercial data mining problems. Elaboration of shortcomings of WordNet confronting with a limited number of classes in text classification is included in this study as well.

4.5 Case study

In order to get familiarized with the position of the company in the industry and the data structure, a clear image on the case study company activities and the way data in question is generated is necessary. Further, we introduce data pre-preparatory steps based on the data attributes and characteristics.

4.5.1 Data

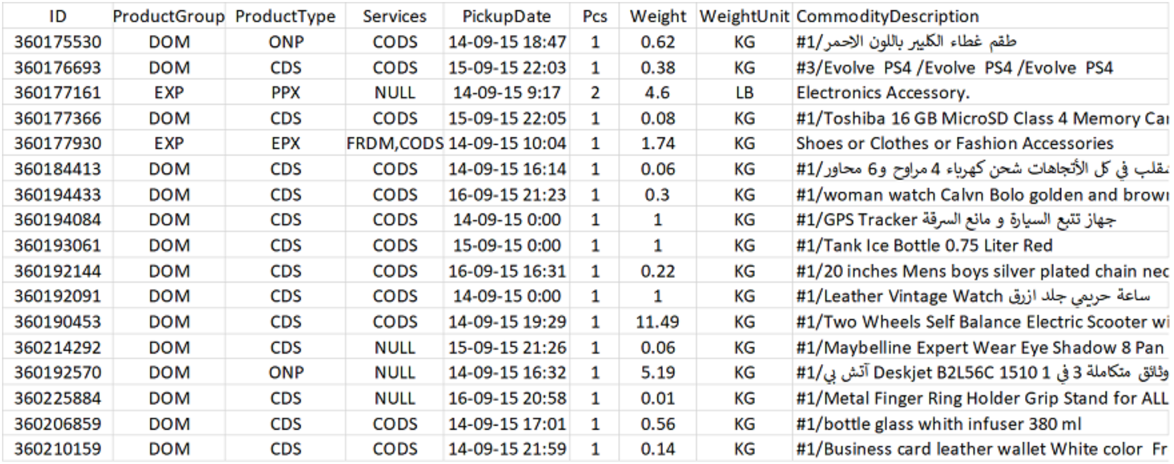
Commodity classification task we have at hand deals with a partial dataset of 100,000 items of commodities delivered to customers by a parcel delivery company. The parcel delivery company has invested in two major delivery systems:

- The first one is an international E-commerce based shopping ship service that aims to connect major merchandising and shopping platforms like eBay and Amazon with potential customers in countries with less product delivery coverage. 250,000 customers have recently used this initiative.
- The second service offered by the company is their domestic parcel delivery system that is generating loads of data regarding package features. Statistics showed that 3 million users have used this service. The parcel delivery company has focused on three main merchants with the highest market shares.

Data attribute types vary from size dimensions and weight to payment type and destination. The dataset has a commodity description attribute which is a very specific introduction of the contents of the parcel.The partial dataset is already classified using the 70 classes defined by the company. The need for a general categorization into 70 pre-defined classes is expressed by the parcel delivery company for this type of data and future data records. Assigning a class to each commodity automatically using text analysis becomes challenging by records in dual languages of English and Arabic. Commodity description attribute provides a range of descriptive properties for each item that introduce the brand, technical specifications, model, etc. As commodity description shares various property names with the original classes, it is reasonable to take it into consideration as the feature extraction source (Figure 4.1). Text pre-processing is to be done in order to normalize the text.

4.5.2 Pre-processing

The main obstacle in normalizing text in this dataset is the fact that in some cases, the commodity description is added in Arabic or it is a combination of Arabic and English. Such text inputs would drastically lower the classifier accuracy as wrong or no specific classes would be assigned to commodities with such descriptions. While dealing with big datasets, one approach could be to remove commodity records with descriptions in Arabic. Our text classification approach however, proposes to translate the descriptions into English. We used the facilities of Google sheets and the Google Translate extension which is able to translate all characters from one desired language to another. One major benefit of this approach is the ability of Google tools suite for processing substantial number of records of data and to detect terms from Arabic inside a mixed commodity description including terms from Arabic and English. Moreover, if the Google Sheets and



ID	ProductGroup	ProductType	Services	PickupDate	Pcs	Weight	WeightUnit	CommodityDescription
360175530	DOM	ONP	CODS	14-09-15 18:47	1	0.62	KG	#1/طقم غطاء الكلير باللون الاحمر
360176693	DOM	CDS	CODS	15-09-15 22:03	1	0.38	KG	#3/Evolve PS4 /Evolve PS4 /Evolve PS4
360177161	EXP	PPX	NULL	14-09-15 9:17	2	4.6	LB	Electronics Accessory.
360177366	DOM	CDS	CODS	15-09-15 22:05	1	0.08	KG	#1/Toshiba 16 GB MicroSD Class 4 Memory Car
360177930	EXP	EPX	FRDM,CODS	14-09-15 10:04	1	1.74	KG	Shoes or Clothes or Fashion Accessories
360184413	DOM	CDS	CODS	14-09-15 16:14	1	0.06	KG	#1/نقلب في كل الأتجاهات شحن كهرباء 4 مراوح و6 محاور
360194433	DOM	CDS	CODS	16-09-15 21:23	1	0.3	KG	#1/woman watch Calvn Bolo golden and browi
360194084	DOM	CDS	CODS	14-09-15 0:00	1	1	KG	#1/GPS Tracker جهاز تتبع السيارة و مانع السرقة
360193061	DOM	CDS	CODS	15-09-15 0:00	1	1	KG	#1/Tank Ice Bottle 0.75 Liter Red
360192144	DOM	CDS	CODS	16-09-15 16:31	1	0.22	KG	#1/20 inches Mens boys silver plated chain nec
360192091	DOM	CDS	CODS	14-09-15 0:00	1	1	KG	#1/Leather Vintage Watch ساعة حريمي جلد ازرق
360190453	DOM	CDS	CODS	14-09-15 19:29	1	11.49	KG	#1/Two Wheels Self Balance Electric Scooter wi
360214292	DOM	CDS	NULL	15-09-15 21:26	1	0.06	KG	#1/Maybelline Expert Wear Eye Shadow 8 Pan
360192570	DOM	ONP	NULL	14-09-15 16:32	1	5.19	KG	#1/آتش بي Deskjet B2L56C 1510 وثائق متكاملة 3 في 1
360225884	DOM	CDS	NULL	16-09-15 20:58	1	0.01	KG	#1/Metal Finger Ring Holder Grip Stand for ALL
360206859	DOM	CDS	CODS	14-09-15 17:01	1	0.56	KG	#1/bottle glass whith infuser 380 ml
360210159	DOM	CDS	CODS	14-09-15 21:59	1	0.14	KG	#1/Business card leather wallet White color Fr

Figure 4.1: A sample of data before pre-processing

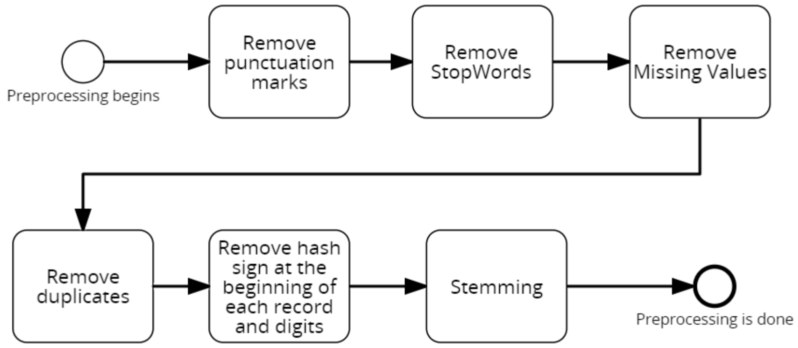


Figure 4.2: Pre-processing steps

Google Translate combination detects a row that is completely in English, it only returns the row to the output file and no modification is performed. Translation function accuracy was assessed as highly precise according to four native Arabic speakers. These native speakers include two university colleagues and two parcel delivery company experts.

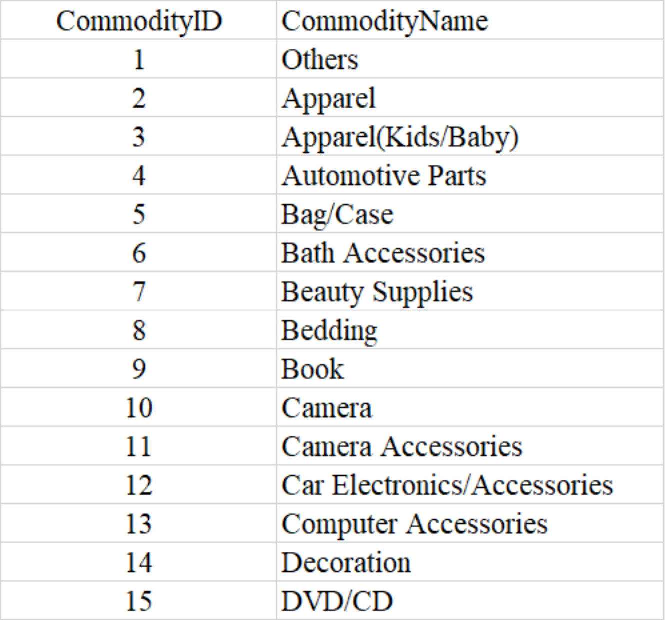
In some cases, commodity description attribute includes terms which represent distinctive classes such as brand names or the exact product category such as *computer* or *perfume*. Keeping such terms is vital based on insight they provide with respect to the pre-defined classes. Removing punctuation marks and English stop words are respectively the next steps in case study, as the focus in item classification is on proper nouns. English stop words removal was done using the pre-defined set of *stopwords.words('english')* from NLTK. This list is available for inspection in appendix B. Numbers and hash signs are as well removed as they will not create added value. Missing values from commodity description column are few but possible. Removing such records is necessary along with duplicate data instances such as *Men sports or casual shorts*, which occurred many times. Further we implemented stemming, using the stemmer based on the *Porter Stemmer* in order to normalize further proper commodity names. The role of a stemmer in text pre-processing as we have stated before, is to reduce the number of derived forms of words and hence, achieving less redundant features. Removing prefixes is the other advantage in this case. For large corpora, stemmers have led to better results [76]. In our case, all forms of *package*, *packages* and *packaging* are children of the root form *pack* for instance. Similarly, stemming significantly impacts the multiple forms of verbs usage. Figure 4.2 is visualizing the schematic of pre-processing steps which were applied in this study. In the next part, we will go through model selection.

4.6 Model selection based on text structure

Natural language processing tasks vary from part of text tokenization to sentiment detection and therefore, multiple probability distribution estimation techniques can be applied to text classification projects. Our desired task is to assign a class to a set of terms using pre-labeled training data. 70 pre-defined classes imply the use of supervised text classification. The algorithm used to assign respective classes requires a set of distinctive classification features for each item as training data, along with a set of known item classes. The machine learning algorithm then models these feature sets and extracts a classification schematic that can be used for incoming data instances. In this cae, the transformation function will be implemented on new data, feature extraction is performed again and the predicted model will output the results [77].

Availability of pre-classified data instances plus the known 70 classes of commodities, is the ground infrastructure of the framework implemented in this study. To perceive the limited role of unsupervised term mapping approaches such as WordNet, a brief introduction can be helpful. Recent use of WordNet has spread in academic text classification use cases due to the availability of a comprehensive network of word trees, creating a hierarchy of terms and term collocations. Such a hierarchy connects less frequent terms with less information on the class with terms that represent distinctive knowledge over a certain class. Initial use of WordNet in text analysis was a step toward building language models using treebanks of synonyms and hyponyms of words. The NLTK package for Python represents a sound implementation of WordNet and has gathered tools to perform various NLP information extraction techniques. Despite huge capabilities of using it, the obstacle on our way is linked with the pre-defined 70 restricted classes. Figure 4.3 visualizes a list with a subset of these predefined classes.

Implementation of WordNet was planned to be based on evaluating the commodity description word by word. Theoretically, our plan was to find all the synonyms for each term in a record. The Look up function would then match if any of the words match any synonyms found in WordNet. If not, we move one level deeper in the word tree and tried again. In this way, we test the possibility of term substitution to reduce the term diversity and their respective classification features even further. While using WordNet is a type of unsupervised machine learning, it can improve the efficiency of a supervised document classification. Still, restricted classes we have in our study did not map completely and precisely onto equivalent WordNet categories in implementation of WordNet in R. For instance, while mapping records with descriptions including a term such as *powerbank* can be done using its respective superior term *accessory* and *mobile accessory* which will be matched with the class *phone accessories* in the 70 classes, in case of occurrence of terms such as *table*, term hierarchy would propose a hyponym superior terms of *furniture* and *office appliance* respectively, which cannot be mapped to any of the available 70 classes. The deficiency of WordNet approach in this study is discussed further in the conclusion section. As a result, contribution of WordNet in



CommodityID	CommodityName
1	Others
2	Apparel
3	Apparel(Kids/Baby)
4	Automotive Parts
5	Bag/Case
6	Bath Accessories
7	Beauty Supplies
8	Bedding
9	Book
10	Camera
11	Camera Accessories
12	Car Electronics/Accessories
13	Computer Accessories
14	Decoration
15	DVD/CD

Figure 4.3: A subset of the 70 predefined classes

this study is not considerable.

A set of classification algorithms were used to train mutiple document classification models. A comparative statistical analysis will show the state-of-the-art modeling capabilities of each classifier. Two popular classification algorithms in this context are Naïve Bayes and K-Nearest Neighbor, due to their predictive capabilities and precise performance. Naïve Bayes will be error-prone for small datasets while it is easy to implement in text categorization problems. However, with regard to classification accuracy it will not be preferred over Support Vector Machines [78, 79]. K-Nearest Neighbor (KNN) requires detection and calculating the K-nearest neigbors distances in the training dataset-and imposes the additional task of selection of the relevant amount for K-, which is rather costly in computation. Additionally, similar class characteristics-attributes which are very similar since their relative classes are similar-or data with big number of features would reduce the performance of KNN. KNN also determines similar weights for features in all classes [80]. For instance, considering two classes of *sad* and *extremely sad* in categorizing a couple of statements on social behavior, terms such as *inconvenient* and *tense* receive the same weights. Although the classes are overlapping and in reallity these two terms should get relatively higher and lower weights in *sad* and *extremely sad* categories respectively. This is the case in many text analysis tasks. Since the concept of Support Vector Machines (SVMs) has been explicitly introduced and used in the next chapters, we have developed models based on classification algorithms which are less appreciated in text classification specific tasks as well to see the effect of current documents nature and structure on their performance. These algorithms include Boosting [81], GLMNET [82], Maximum Entropy (MAXENT) [83] and Supervised Latent Dirichlet Allocation (SLDA) [84].

Kudo et al. [85] have introduced a framework based on Boosting algorithm to classify semi-structured sentences represented as a labeled ordered tree. The paper presents the sub-trees as features extracted from text. Boosting algorithms leverage the weak learners, classifiers which predict the right class only slightly better than random guessing, to become strong learners, which would correlate rigorously with the right classification. Each classifier will be trained based on the hardest instances to classify by previous classifier [86].

A modified regression model capable of handling linear, logistic and multinomial regression is developed in GLMNET package in R. Some advantages of this model is the ability to work with big number of features, cross validation and allowance for sparse matrices which will be effective in text mining tasks.

Maximum Entropy or multinomial logistic regression proposes a set of derived constraints from the labeled data to define a class-level expectation. In other words, it restricts the number of features for each class by a set of constraints, which makes this algorithm efficient for text classification [87].

In supervised LDA (SLDA), each document is accompanied by a response variable, such as the case of the number of times a document is downloaded from the web. Then the response variable is the number of downloads. *Topic Modeling* which is the ground concept of LDA, proposes that each document is represented as a discrete set of random variables which are the composing words. Then each document is composed of a set of words which are in the framework of a latent topic-or an unknown distribution over the terms-. In this explanation, a set of documents includes N topics, while each document inside the set is formed by a distribution of a number of topics only [84].

Implementation for all four algorithms is available using R packages. MAXENT classifier, has proved to be efficient in text categorization scope. Della Pietra et al. [88] state that given a set of classes C with

$$C : \{c_1, c_2, ..., c_N\} \qquad (4.1)$$

A set of labeled documents with classes

$$\{(d_1, c_1)(d_2, c_2)(d_3, c_3), ..., (d_n, c_N)\} \qquad (4.2)$$

For the document d and the pertinent class c, MAXENT representation of $P(c \mid d)$ is an exponential form of:

$$P(c \mid d) := \frac{1}{Z(d)} \exp(\sum_i \lambda_{i,c} F_{i,c}(d,c)) \qquad (4.3)$$

Here, $\lambda_{i,c}$ is the weight parameter assigned to each feature. i and c respectively represent the index given to set of features (terms) and classes. $Z(d)$ is the normalization constant which adjusts the probability distribution:

$$Z(d) = \sum_c \exp(\sum_i \lambda_{i,c} F_{i,c}(d,c)) \qquad (4.4)$$

$F_{i,c}$ is a binary function and the output for each instance to be classified is either zero or one [89]. Refer to equation 4.5. The function assumes a context which distinguishes a class. Consider an example. We want to know how the term *helping* in the phrase *"the helping hand"* should be tagged as a POS tag. Maximum Entropy begins by the least informative assumption and grows grasping the real context by observing occurring instances. For instance, in the example of rolling a dice, the initial assumption is that all faces of the dice are equal and then each face can occur with a probability of 1/6. The initial assumption for the next dice rolling would then also be 1/6. Further, if it is observed that next roll dices tend to take a specific amount, the probability of next roll dices would change. In our text example, based on the previous context-which is the training data that has trained a classifier-, the classifier might decide that *helping* is assigned a POS tag of *VBG* as a verb, since training data shows that in 30% of cases where *helping* has occurred previous to a noun, it received the *VBG* tag. More and more of similar constraints might apply, as the number of classes or features increases. $F_{i,c}$ is basically the function that represent these set of constraints that apply in knwoledge extraction. Yet, by training more and more instances and observing the fact that the actual class changes based on the context-e.g. the next term-, the classifier might assign different POS tag to the term *helping.* The reason lies in the principle of the Maximum Entropy algorithm, stating that the probability of assigning a certain class tag to a document given a specific context must maximize the entropy of the classification system. No biases in this way are introduced in the classification system, according to [90].

$$F_{i,c} = \begin{cases} 1 & \text{if} \quad \text{the term is relevant to context} \quad \textit{and} \quad \text{class is true} \\ 0 & \text{if} \quad otherwise \end{cases} \qquad (4.5)$$

Generally speaking, features in Maximum Entropy model are a function of classes predicted and binary context constraints. In [91], the author has introduced the representation of such a function in his part-of-speech tagger. A feature in this explanation is a set of yes/no questions with a constraint applied to each word. Any word that satisfies the constraint would be a feature. Maximum entropy models state the fact that among all probability distributions available to model testable data, which in this study is the groupings of the commodities, the one model with the highest entropy is the true model. The main substantial feature in maximum entropy in this study is the ability to handle comprehensive features which will be the case here. Document term matrix or feature matrix is created using the package tm of R, and it is the nominal representation of the most distinctive terms used in each class for categorization. Additionally, a feature-cutoff will leave features occurring less than a specific number of times. Figure 4.4 represents an overview of model generation procedure in this study.

In this experiment, we have employed the RTextTools library of R programming environment for machine learning in text classification [92]. The R distribution of MAXENT, trains a maximum entropy model using a document term matrix and feature vector. Document term matrices are numerical representations of documents. Each document would be represented word by word in rows and the columns state the existence or absence of words in each document. A feature vector would as well represent the given labels. Eventually, the trained model is tested on a portion of the initial data set to test the performance of the model.

To train models, the experts primarily count on formal language document datasets. Of course, in case of assessing informal or slang orientation or classification, the use of these datasets is limited and inefficient compared to cases such as topic based classification of news. Examples of formal language based document banks include headlines of New York Times [93] and bills from United States Congress. In this study though, to achieve the highest degree of accuracy, we have used a pre-labeled subset of the complete dataset presented by the parcel delivery company, to ensure efficient feature selection and extraction for training classification models. Additionally, we ensured the same pre-processing techniques to be used on both train and test dataset. For instance, stemming was done on both cases, to prevent any feature inconsistency among the two sets. For all classification algorithms in this experiment, we used the default parameters specified by the R package RTextTools.

4.7 Discussion and results

Using stratified sampling and with respect to 70 classes at hand, we used 80% of data-around 80,000 records-from each category for training and 20%-the remaining 20,000 records-for testing instead of random sampling. Such a task would normalize the effect of all categories on model training and ensures the occurrence of samples from each category. The split ratio is rather not arbitrary in machine learning tasks. It is necessary to note that maximizing the amount for training data would enhance the model performance. Yet, it depends on the task and how

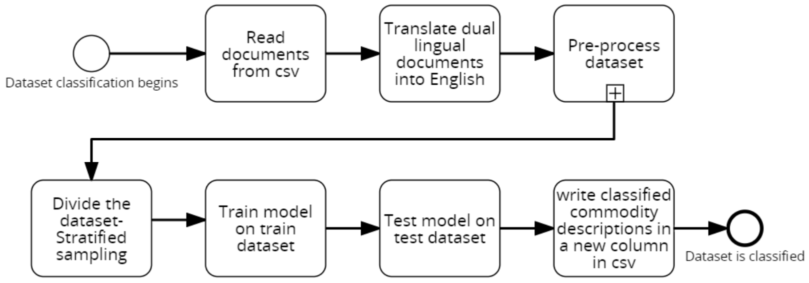

Dataset classification begins
Read documents from csv
Translate dual lingual documents into English
Pre-process dataset
Divide the dataset- Stratified sampling
Train model on train dataset
Test model on test dataset
write classified commodity descriptions in a new column in csv
Dataset is classified

Figure 4.4: Model generation steps

much data the model would require for training and testing phases [94]. For instance, if millions of data records are available, even more than 95% of data can be assigned for training. Still, split ratios such as 70:30(train-test), 80:20(train-test) or even 60:20:20(train-test-validation) are known to be of good practice in smaller datasets. Data loaded will be used to create a document-term matrix. Pre-processing options are available in this step. Using Tf-Idf, a set of rare but significant terms are selected to train classification models. Training models based on a couple of classification algorithms will be initiated afterwards. Trained models are applied on the test dataset based on the trained models. Finally, the analytics for classification task will be provided for evaluation of the approach. Eventually, the results will be exported to the output file desired by the user.

The MAXENT classification model was applied on the test dataset and achieved the precision and recall measures of 0.9365217 and 0.9144928, respectively. Precision measure introduces the number of right returned instances that are queried by the classifier.

$$Precision = \frac{TruePositive(t_p)}{TruePositive(t_p) + FalsePositive(f_p)} \tag{4.6}$$

In this equation, the true positives and false positives are respectively the number of hits or true instances to be found and the number of instances that are selected as hit but belong to other groups. In other words, precision detects the percentage of relevant items selected. Recall is the other performance measure for classification tasks and it detects the percentage of selected items that are relevant.

$$Recall = \frac{TruePositive(t_p)}{TruePositive(t_p) + FalseNegative(f_n)} \tag{4.7}$$

False negative in this equation indicates the case where an instance is rejected to be in the respective class, while it actually belongs to the class. Precision and recall are both indications of the relevancy of model. Table 4.1 shows precision, recall, and F-score measures for the five classification algorithms applied on our test dataset. In addition to the aforementioned classification algorithms, a model trained based on the Support Vector Machine is as well provided for comparison purposes. Vector representation of text features results in sparse matrices as stated before. For SVM, without fine-tuning the decision boundary parameter or using the default settings in R and Python packages, the results might not necessarily be desirable. Further elaboration is presented in chapter 5.

Algorithm	Precision	Recall	F-score
GLMNET	0.5207246	0.4208696	0.4407246
MAXENT	0.9365217	0.9144928	0.9228986
BOOSTING	0.9502899	0.9165217	0.9260870
SLDA	0.8911594	0.8952174	0.8839130
SVM	0.9026087	0.8834783	0.8882609

Table 4.1: Precision, Recall and F-score for each model applied on test dataset

The classification model was tested on the test dataset, considering keeping only the same terms presented in training dataset. Limiting the term set would prevent the model to face unknown terms and features respectively. A confidence level for each output is calculated as well along with class prediction which will be compared with a certain threshold for level of confidence. If each record has a lower than threshold confidence level-0.5 in this study-, it is categorized as *others.*

Some of the models that have been tested in this study have showed close precision and recall measures. With respect to precision, Boosting, MAXENT, SVM and SLDA have gained the highest ranks respectively, outperforming GLMNET significantly with the precision roughly around 52%. Recall measure for GLMNET is low as well compared to the rest of the models. Big number of classes in our case study lowers the results for GLMNENT classifier. Furthermore, GLMNET is a low-memory classifier. Yet it is able to handle substantial in size sparse matrices of features, which is applicable in this case study. MAXENT and BOOSTING gain very close recall measures, followed by SLDA which would outperform SVM by around 1%. Figure 4.5 and figure 4.6 represent the visualization for precision and recall achievements for all models tested in this study.

The F-score for each classification algorithm is illustrated in table 4.1. The F-score is simply the harmonic mean of precision and recall. For the case where precision and recall are both one, we multiply the measure by two.

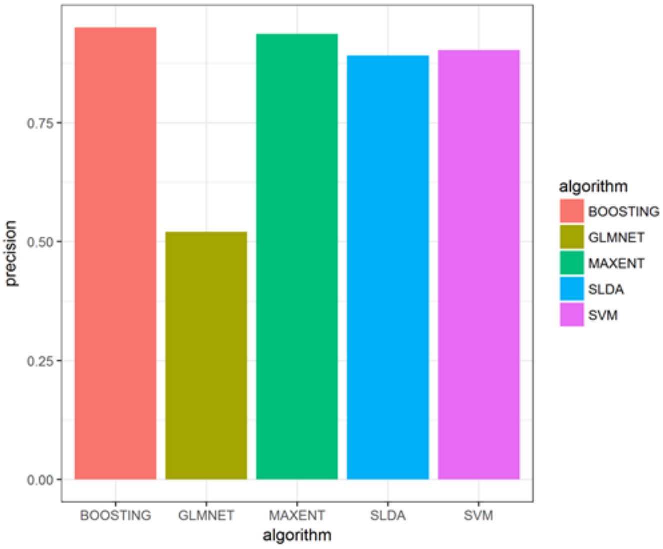


Figure 4.5: Precision measure comparison charts for all models

$$F - score = \frac{Precision \times Recall}{Precision + Recall} \tag{4.8}$$

The highest F-score is achieved by BOOSTING and MAXENT respectively, followed by SVM and as expected, the F-score for GLMNET is considerably lower, around 44%. Figure 4.7 represents the comparison of F-score achieved by all models in this study.

The commodity category of *others* include low confidence labels from the commodity description records. These instances would not receive a class tag or have loose bonds with any of the classes. In other words, the feature set extracted from these description items is by no means clearly showing pointing to any of the 70 classes defined.

The implementation of MAXENT classifier in R has proved to be memory friendly and showed considerable precision. Hence, MAXENT is the selected algorithm for the output classification schema we present. Models for MAXENT and GLMNET can run on local hosts while the rest of the models can run out of memory even on our access point to Flemish Supercomputer that is a powerful computing node, which in fact originates from feature diversity and high dimensional vector matrices. Figure 4.8 presents a portion of the output with classified

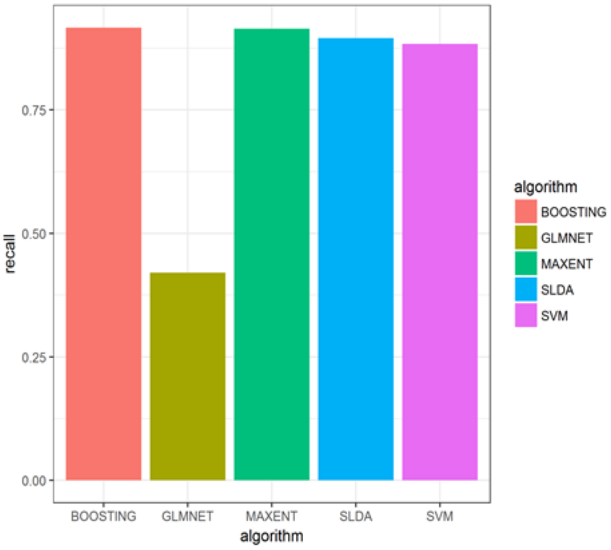


Figure 4.6: Recall measure comparison charts for all models

records. Notice the use of a unified language in commodity descriptions.

4.8 Conclusion and future work

Supervised machine learning has proved to be significantly comprehensive in natural language processing tasks. In one particular case, we have proposed a text classification methodology, specifically used in shipped commodity descriptive datasets. Text including dual language characters makes it challenging to get insights into text classes. Our proposed framework employs a dual language translation function to convert every item in a combination of Arabic and English text strings in our dataset into English. Several pre-processing and data cleansing techniques are implemented to prepare the data to be fed into a model based on a couple of supervised machine learning algorithms. The case study introduced in this research is a typical data analysis application that many companies in production and service industry face daily. As long as descriptive data stream is loaded in databases representing broad aspects of products, we are able to handle dual-lingual text, remove noise and classify each item respectively. We have addressed the problem for the parcel delivery company to categorize items and answer socio-demographics-based questions concerning customers and how their parcel delivery records can be monetized as valuable data source for merchandising rivals. Questions such as ”In which top 5 countries the highest number of jewelry buyers live?” or ”How long did it take for parcels including electronics with chargeable batteries

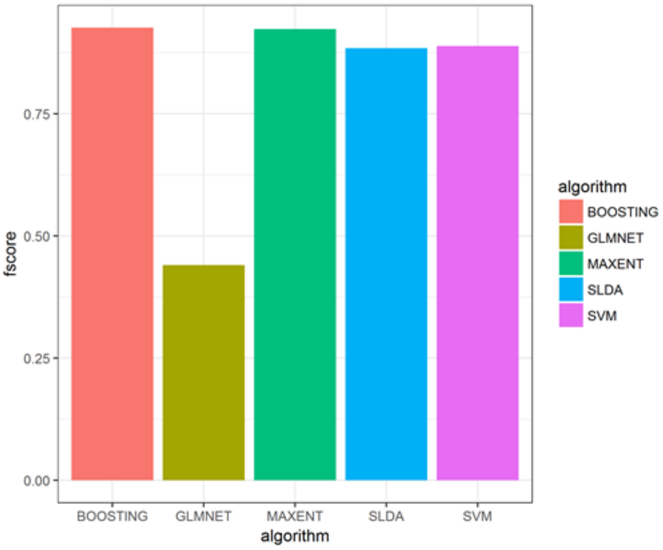


Figure 4.7: F-score measure comparison for all models

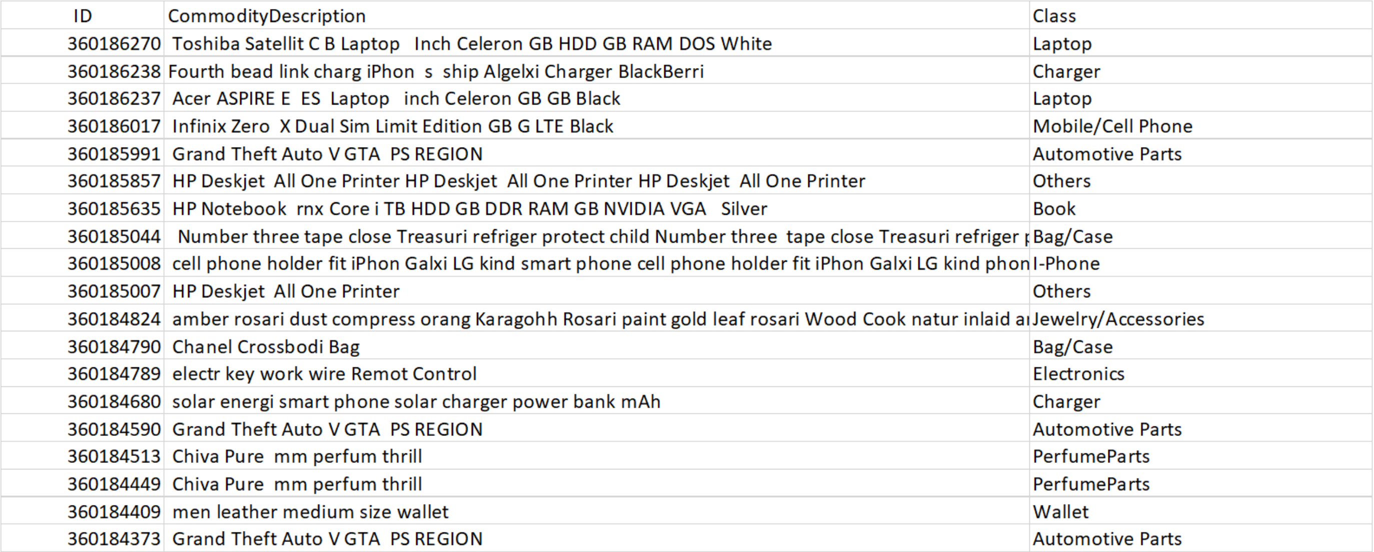
shipped from UAE reach Qatar?" can be answered using the proposed commodity classification system. Answering such questions would enhance the potentials of the parcel delivery company in stabilizing its position as the most reliable delivery partner for certain important or highly valuable commodities. Moreover, marketing partners will be interesting in knowing which regions, age groups or customer gender types are interested in which types of consumer products. Such knowledge can reinforce marketing campaigns. We have gained notable precision and recall measures, proving our methodology to be responsive and accurate.

Yet, there are practical obstacles on application of WordNet in this study. As a supervised machine learning case study, the use of WordNet is limited and it resulted in complications in analysis of the results. First, the input should be assessed based on the coverage and semantic based limitations of WordNet. Our dataset includes thousands of terms forming each commodity description. Many of these terms do not exist in WordNet since WordNet was initially released as a hand-coded thesaurus and is not necessarily intended to be used for heavy NLP and classification tasks. For instance, a certain class in the 70 categories we have at hand is *Apparel*. Yet, many data records include terms such as *coat* and *hanger* which point towards an accessory. But from a uni-gram feature viewpoint, this commodity is either a type of *apparel* or something else which is either tagged as *others* or an irrelevant class, depending on the effect of the similar terms in training set on the model. WordNet does not detect the most basic lexical entities as well. For instance, word combinations such as *Electronic Circuit* might be detected as a hyponym of *geography* or *shapes* because of *circuit* instead of *electronics*. In other words, the control over what the WordNet would propose in word mapping depends on various elements. The same case would result in wrong category proposition, when the limited number of categories we have for classification is much smaller than what WordNet is able to predict, and this would lead to wrong predictions or inability of term mapping.

From a practical point of view, the business partner is able to monetize such a categorization function with respect to the highly desirable items specifically in online shopping. In case the vendor, the value and the frequency of sale for each commodity is available, online markets will gain more insights into sale and marketing. Other example applications would be in geo-tagging parcel delivery service selection with respect to category and value of parcel.

While our solution to the commodity classification problem is based on supervised machine learning, further insight into customization based on WordNet to make it proper for similar cases and implementing the N-gram model can be further studied and the results could be interesting.

The classification algorithms presented in this chapter showed a considerable performance in dealing with relatively limited number of features. To fill the gap for classification of documents with bigger sets of features, during the next chapter, Support Vector Machine will be elaborated as an effective text classification algorithm with vast sets of features to provide the fundamental knowledge in interpretation of methodology necessary to answer to the third challenge of this study.



ID	CommodityDescription	Class
360186270	Toshiba Satellit C B Laptop Inch Celeron GB HDD GB RAM DOS White	Laptop
360186238	Fourth bead link charg iPhon s ship Algelxi Charger BlackBerri	Charger
360186237	Acer ASPIRE E ES Laptop inch Celeron GB GB Black	Laptop
360186017	Infinix Zero X Dual Sim Limit Edition GB G LTE Black	Mobile/Cell Phone
360185991	Grand Theft Auto V GTA PS REGION	Automotive Parts
360185857	HP Deskjet All One Printer HP Deskjet All One Printer HP Deskjet All One Printer	Others
360185635	HP Notebook rnx Core i TB HDD GB DDR RAM GB NVIDIA VGA Silver	Book
360185044	Number three tape close Treasuri refriger protect child Number three tape close Treasuri refriger p	Bag/Case
360185008	cell phone holder fit iPhon Galxi LG kind smart phone cell phone holder fit iPhon Galxi LG kind phon	I-Phone
360185007	HP Deskjet All One Printer	Others
360184824	amber rosari dust compress orang Karagohh Rosari paint gold leaf rosari Wood Cook natur inlaid a	Jewelry/Accessories
360184790	Chanel Crossbodi Bag	Bag/Case
360184789	electr key work wire Remot Control	Electronics
360184680	solar energi smart phone solar charger power bank mAh	Charger
360184590	Grand Theft Auto V GTA PS REGION	Automotive Parts
360184513	Chiva Pure mm perfum thrill	PerfumeParts
360184449	Chiva Pure mm perfum thrill	PerfumeParts
360184409	men leather medium size wallet	Wallet
360184373	Grand Theft Auto V GTA PS REGION	Automotive Parts

Figure 4.8: A subset of output file with classified samples

Chapter 5

From Vector Space Model to Support Vector Machine

In order to perceive a detailed background on the application framework for text classification represented in chapter 6, having a clear view of support vector model methodology is vital. This chapter has selected a widely used numerical feature representation methodology. We have explored the fundamentals of vector space model and facilities supported by support vector machines.

5.1 Documents as vectors

In previous chapter, we elaborated a text classification problem to address the need for a commodity classification scheme that can enable a parcel delivery company with insight over their actions and the potential to convert this insight to added value. The documents in that case study were relatively small in terms of the number of terms. For more complex document structures and bigger documents, we further investigate the concept of a *Vector Space*, in which each vector represents a document. To answer the questions in resolving the third challenge of the research motivation, the need for a more sophisticated classification schema that can capture hidden patterns in documents is inevitable. We believe that revealing patterns beyond term level is effective in classification of documents which share a solid structure such as rules. In this respect, features are formed from other properties of text strings other than their physical representation, such as terms part-of-speech tags.

Support Vector Network, initially introduced by Cortes and Vapnik [95], formulates a pattern recognition framework using a feature presentation of n dimensions and a non-linear mapping of input vectors. As it was described in the previous chapter, data cleaning techniques return a set of unstructured text strings as proper documents for vector space model. Using a statistical measure of significance such as Tf-Idf, a weighting schema is shaped to prune a high dimensional set of features. Hence, only a specific set of features, which are ranked higher than a threshold are kept for document representation. Finally, we can assume a set of documents with respect to ranking score ready for vector space model.

In Vector Space Model, terms of documents are axes of the space, and as in text documents we face a lot of terms as features, we are dealing with a hyper-space, with any number of dimensions. A sense of similarity, framed in the form a proximity measure among documents, determines relatedness of documents, and hence defines profiles and classes that can categorize documents. Consider two documents of d_1 and d_2, each visualizing the following statements, denoted with $\vec{d_1}$ and $\vec{d_2}$:

$$\vec{d_1} = (t_{1,1}, t_{2,1}, ..., t_{n,1})$$

and

$$\vec{d_2} = (t_{1,2}, t_{2,2}, ..., t_{m,2})$$

where t shows certain terms used in each document. Assume document d_1 to be *renaissance heritage specialist* and d_2 to be *contemporary art observatory.* In d_1 for instance, $t_{1,1}$ is *renaissance* and $t_{1,2}$ is *heritage.* As a result, in a hypothetical vector space, corresponding vectors $\vec{d_1}$ and $\vec{d_2}$ visualized in figure 5.1, respectively show a close proximity measure to *classic* and *modern* categories, as terms occurred in d_1 and d_2 have a short distance with exclusive terms in two classes of *classic* and *modern.*

Looking back at figure 5.1, in a two dimensional vector space with two axes of *classic* and *modern*, $\vec{d_1}$ terms tend to lean toward the classic axis, due to higher weights for terms *renaissance* and *heritage.* These terms are nodes of a cognitive map in a full hypernymy network of the word classic. Similarly, in $\vec{d_2}$, term weighting results in a close proximity measure to modern axis. It is important to understand that different proximity measures can be defined in a vector space. One is the use of *Euclidean Distance* between each vector and the reference vectors. Given the two vectors of assumption $(\vec{d_1}, \vec{d_2}) \in \Re^n$, Euclidean distance of $\vec{d_1}$ and $\vec{d}_{classic}$ is defined as:

$$d(\vec{d_1}, \vec{d}_{classic}) = \left\| \vec{d_1} - \vec{d}_{classic} \right\| = \sqrt{(d_{1_1} - d_{classic_1})^2 + (d_{1_2} - d_{classic_2})^2 + ... + (d_{1_n} - d_{classic_n})^2} \tag{5.1}$$

Where $d_{1,1}, d_{1,2}, ..., d_{1,n}$ are specific terms in document d_1 and similarly, representing terms of $d_{classic}$ are denoted using $d_{classic_1}, d_{classic_2}, ..., d_{classic_n}$. Euclidean

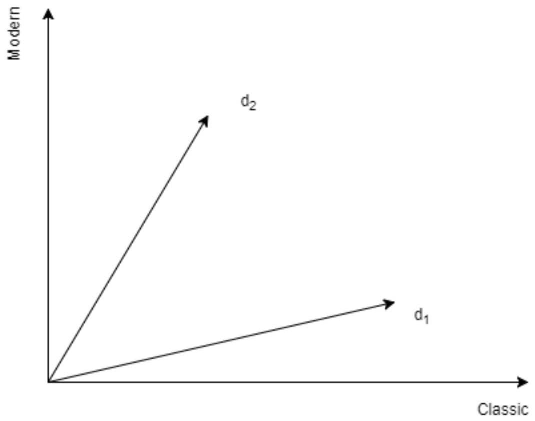


Figure 5.1: Vector space model for d_1 and d_2

distance is widely used in text mining applications, specifically in text clustering [96]. A second similarity measure is *Cosine Similarity.* It is based on a basic notation of the visual distance that the eye feels from the already categorized vectors. Consider a new input document query, which is denoted as follows:

$$q = (t_{1,q}, t_{2,q}, ..., t_{n,q}) \tag{5.2}$$

The angle between the vector d_1 for instance, and d_2 is bigger than the angle with d_1. As a result, reflecting the image of q on each vector d_1 and d_2 returns an absolute amount that can be determined as a measure of similarity. Cosine of the angle is this case is of more sense to calculate. Consider figure 5.2 and equation 5.3.

In order to determine the cosine similarity, we assume documents d_1 and d_2 are already in training data, given respective classes of classic and modern. Given two documents, in this case the cosine similarity of q and d_1 or d_2 can be calculated as follows. Higher cosine similarity, implies a higher degree of similarity among documents.

$$\cos_{similarity}(\vec{q}, \vec{t_1}) = \frac{\vec{q} \cdot \vec{d_1}}{\|\vec{q}\| \cdot \left\|\vec{d_1}\right\|} \tag{5.3}$$

Consider a duplicate copy of document d_1, which is appended to d_1, resulting in a document with similar terms and double length compared to d_1 (Figure 5.3).

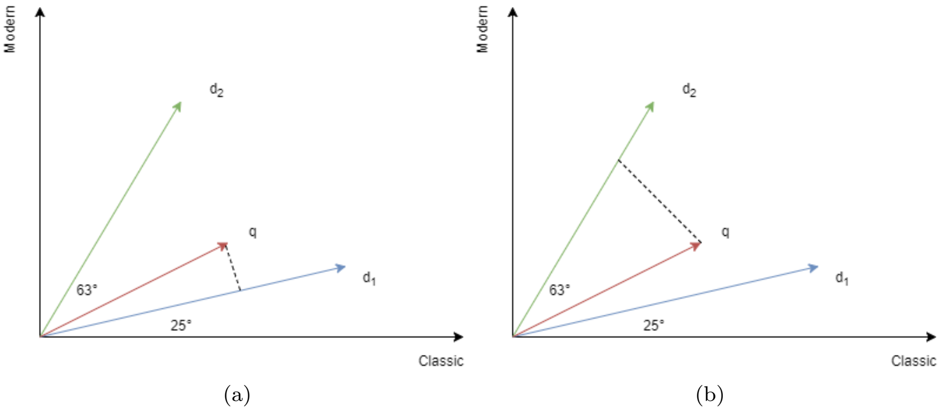


Figure 5.2: Cosine similarity of query q compared with d_1 and d_2

We call this document $d_1{}'$. Now with a new input query q', an identical cosine similarity with d_1 and $d_1{}'$ will be at hand for q'. In other words, documents with the same terms and different total lengths will be treated similar [96].

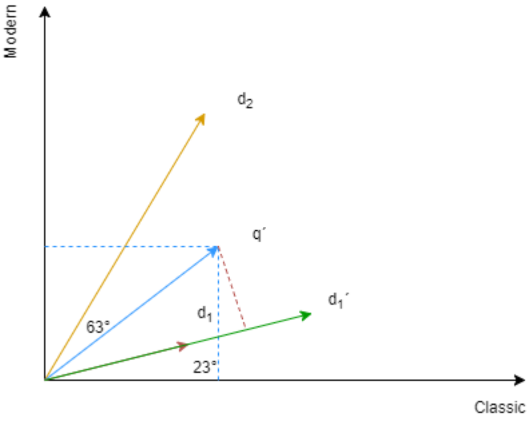


Figure 5.3: Double sized d_1 and the effect on cosine similarity

With a good image over Vector Space Model in mind for a set of documents, we are now able to explore a supervised machine learning approach for text classification built upon this phenomenon. In the next section of this chapter, we review the principles of *support vector machine* text classification algorithm.

5.2 Support Vector Machine

As the name suggests, Support Vector Machine (SVM) inherits the principles of Vector Space Model and is developed from two class entity classification. Considering a number of training data points in a two-dimensional surface-a plane-, these points can be viewed as vectors with the start point of the intersection of two dimensions and the end point of the data point coordinates. In this regard, if the data points are linearly separable, the linear decision boundary is the classifier. Some data points in this case are closest to the decision line. SVM aims to find the decision line in a way that the distance with data points from the two classes closest to the line is maximized [97]. A visualization of such a case is represented in figure 5.4a.

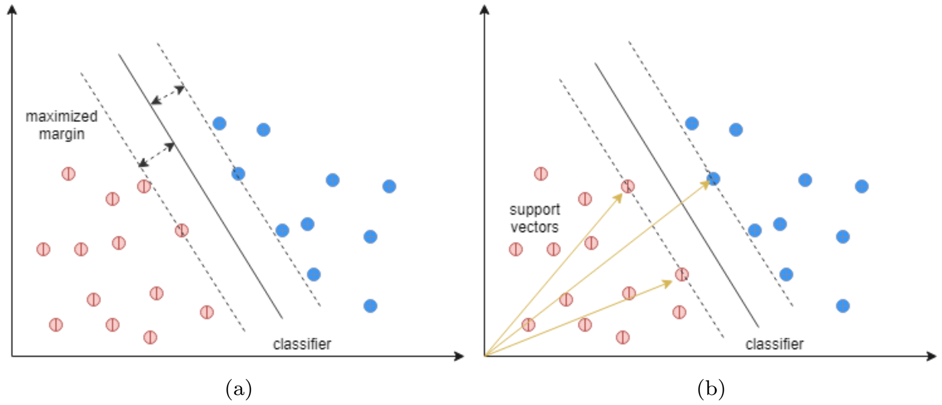


Figure 5.4: Classifier line in two class linear SVM (a) and support vector concept in SVM (b)

By this definition, vectors shaped by the points situated on the margin of the furthest distance with classifier are called the support vectors. Figure 5.4b represents support vectors in the vector space model.

While an increase in dimensions over a text classification task is inevitable due to a substantial number of features, a linear decision boundary-or classifier-is not able to determine the maximum margin of distance between support vectors and the classifier. Such a restriction rules for a plane in $\Re^n$. This means hyperplanes classify documents with multiple features. A hyperplane is a $n-1$ space in $\Re^n$. SVM can be formulated according to [74]. Given an intercept term b and a weight vector $\vec{w}$ which is actually a decision hyperplane normal vector and perpendicular to the hyperplane, all points of $\vec{x}$ satisfy the following equation, having specified the term b:

$$\vec{w} \cdot \vec{x} = -b \qquad (5.4)$$

Now considering a set of training points $D \in (\vec{x}_i, y_i)$, where each member is a point in $\vec{x}$ and y_i the corresponding class, the linear classifier can be defined as:

$$h(\vec{x}) = sign(\vec{w} \cdot \vec{x} + b) \qquad where \quad y_i \in \{+1, -1\} \qquad (5.5)$$

When data points are linearly separable in D, SVM finds the hyperplane with the highest Euclidean distance to the closest data samples to the hyperplane. Such a distance is known as *the margin* and it is shown by δ. Using the aforementioned definition, for n_{th} point on $\vec{x}$ and having a hypothetical hyperplane $< \vec{w}, b >$, the *functional margin* is $y_i(\vec{w} \cdot \vec{x}_i + b)$ and as a result, the total functional margin-or *geometric margin*-of a dataset is twice the size of functional margin for each data point in dataset, as we sum the amount for both classes of +1 and -1. Figure 5.5 represents an example of such a case.

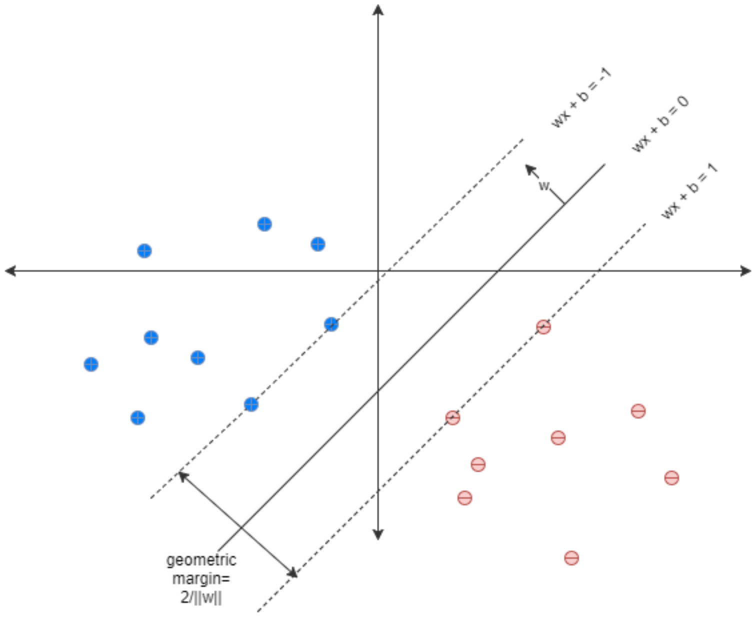


Figure 5.5: Geometric margin in SVM

If we denote the Euclidean distance between a support vector and a certain point $\vec{x}'_i$ on $\vec{x}$ by r and the total functional margin-geometric margin-by ρ, a perpendicular vector will represent this Euclidean distance, as a coefficient of unit vector $\frac{\vec{w}}{\|\vec{w}\|}$ because of a their equal angle with the vertical reference vector. As a result the coordinates of $\vec{x}'_i$ will be:

$$\vec{x}'_i = \vec{x} - yr\frac{\vec{w}}{\|\vec{w}\|} \qquad (5.6)$$

substituting $\vec{x}'_i$ for $\vec{x}$ in $h(\vec{x}) = sign(\vec{w} \cdot \vec{x} + b)$ results in:

$$\vec{w}(\vec{x} - yr\frac{\vec{w}}{\|\vec{w}\|}) + b = 0 \qquad (5.7)$$

So we can calculate r as :

$$r = y\frac{\vec{w} \cdot \vec{x} + b}{\|\vec{w}\|} \qquad (5.8)$$

As stated previously, the geometric margin can be altered but it is still a function of alterations of both $\vec{w}$ and b. Multiplying both the coefficient and the intercept by a specific number still returns the same amount for r. This scaling property of r facilitates finding large number of SVMs. Since, at least one support vector is required to determine $\vec{x}$ and r and we can simplify the functional margin of each point on $\vec{x}$ to be 1, we can alter the formula for all training data as $y_i(\vec{w} \cdot \vec{x} + b) \geq 1$

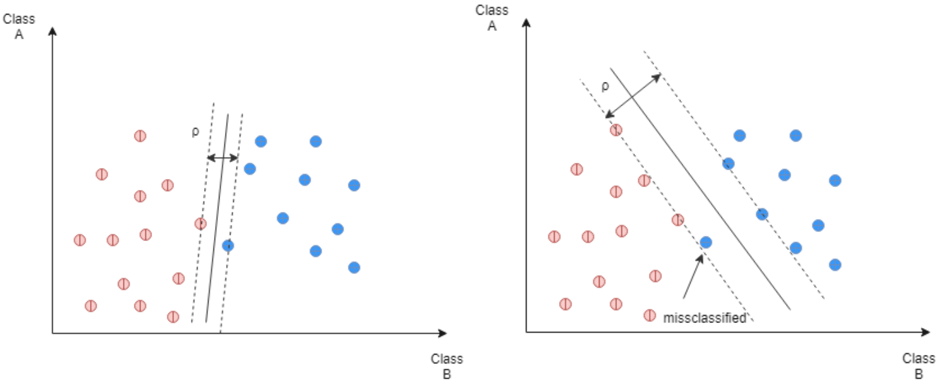
We normalize functional margin as shown in figure 5.5 where $\vec{w} \cdot \vec{x} + b = \pm 1$ and hence the geometric function, twice the size of functional margin is:

$$\frac{\vec{w}}{\|\vec{w}\|}(x_{positive} - x_{negative}) = \frac{2}{\|\vec{w}\|} \qquad (5.9)$$

Since the classifier hyperplane is supposed to be on the furthest possible distance from the support vectors, our goal is to maximize the amount for $\rho = \frac{2}{\|\vec{w}\|}$ or minimizing $\frac{\|\vec{w}\|}{2}$. Hence we have formulated the SVM as two constraints on finding $\vec{w}$ and b:

$$\begin{cases} minimize \quad \frac{1}{2}\vec{w}^T\vec{w} \\ y_i(\vec{w}\cdot\vec{x}_i + b) \geq 1 \quad \forall\{(\vec{x}_i, y_i)\} \in D \end{cases} \tag{5.10}$$

Optimizing the constraint problem can be basically imagined as a trade off problem between the geometric margin and the number of data samples which can be wrongly classified. As shown in figure 5.6.



(a) (b)

Figure 5.6: Trade off between small functional/geometric margin (a) and miss-classified sample(b)

Here, two adjacent parameters to the support vector machine are of great importance. The first one is called γ parameter which basically determines if the classifier is going through all samples correctly. Defining a high value for γ is translated as assigning low significance determination factor to points which are further from the support vectors. As a result, high values for γ can lead to a curvy classifiers which can cover samples deeply inside the irrelevant class. On the other hand low values of γ lead to more straight classifiers, giving more weights of significance to further samples to decision boundary. The other parameter to take into consideration is specified as c and corresponds with the degree of trade-off between accepting wrong classified samples and higher values of ρ or vice versa. Small c values correspond with bigger functional margin and as long as c gets bigger, the margin tends to a lower value and hence, lower chance of miss-classified samples. Having the two parameters at hand, we can finally formulate the SVM for classification task as minimizing the trade-off function:

$$\frac{1}{2}\vec{w}^T\vec{w} + c\sum_{i=1}^{N}\gamma_i \quad where \quad y_i(\vec{w}^T \cdot x_i + b) \geq 1 - \gamma \quad \forall i \in (1, 2, ...n) \tag{5.11}$$

To summarize the task of classification using SVM, the training dataset points are distributed and a decision boundary or classifying hyperplane performs the classification task. This hyperplane is calculated through the aforementioned optimization of error and SVM formula in equation 4.4.

With a clear view over classification task using SVM, we are now able to explore through further machine learning tasks using SVM, such as introducing new feature selection methods to the SVM for text classification purpose, as discussed in the next chapter. During the next section, a discussion over the applicability of SVM on text classification is presented.

5.3 Discussion

To classify documents with similar grammatical structures into a number of classes, SVM would provide admirable learning and modeling capabilities. Two main advantages of SVM in this regard, are the ability to handle sparse matrices of features due to high volume of memory it provides and the fact that almost all document classifications are separable by linear classifiers [21]. Still, the intention is to improve the feature representation by moving from term level features to more semantics based feature levels. The combination of a more developed feature extraction methodology and support vector machine would potentially handle the task of document classification with known language structures.

As a fundamental step towards extracting decision models from documents, which is the answer to the third challenge in this study, the key to success is on extracting a manageable set of classes and classification features. The primary property of a decision model expressed in a document is the co-occurrence of a constraint or condition and an action which is triggered based on the condition. In other words, the structure of these two entities specifies a type of rule or regulation. Consistency levels of structure can be determined in this case, as a means of facilitation in document analysis. This means that documents which share common structure, can use this solid attribute as their identifier-or class label-. The need for a function to capture these repetitive patterns is inevitable in this regard.

Capturing the patterns is the initial key to resolving classification of documents. Yet, it is in our interest to extend the methodology to capture the highest level of lexical-or grammatical-patterns. This would result in better prediction capabilities of the models by determination of the maximum amount of evident or hidden patterns.

We will implement this methodology in the next chapter using the combination of two document term properties: their relative part of speech tags and the N-grams generated from both the terms and the POS tags.

5.4 Conclusion

In this chapter, a detailed study on support vector machines and their capabilities in document categorization was conducted. We elaborated details on linear and non-linear classification task using SVM. Integrated implementation of SVM and NLP techniques in Python has motivated us to detect the solid grammatical structures of specific document sets, and use this aspect in finding relevant classes for next documents. The experiment in chapter 6 provides more information in this regard.

Chapter 6

Hybrid feature extraction for regulatory based documents

In this chapter, transformation functions for modeling text classification algorithms based on Support Vector Machines, as discussed in detail in previous chapter, have been illustrated. In order to elaborate the concepts and frameworks in chapter 5, an experiment with a rule-based document type, which is widely used in decision making systems, has been designed and elaborated.

6.1 Introduction

Text classification is one of the most celebrated applications of text mining in daily business. Recent research shows that text classification forms 18% of features on text mining tools. With that score, it stands on the second place after text analytics with 42% share [98]. Figure 6.1 visually represents the distribution of text mining tools [98].

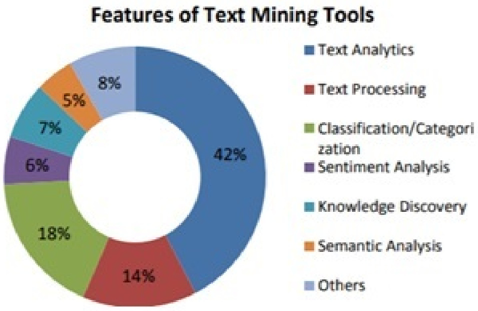


Figure 6.1: Features of Text mining tools [98]

Different units in business are providing and consuming text data every day. Legal departments of every company or organization have the responsibility of creating, modifying and removing various types of databases which express regulations regarding all activities that run a business. These regulations define legal entities and articles, possible situations and actions during each activity. Legal databases which maintain these regulations include thousands of rules, which requires a retrieval system that provides access to a certain article whenever it is needed. In other words, legal departments possess many text documents that need classification, so they can find and use any relevant information at the shortest possible time.

In this study, we aim for an application field in text classification for a specific rule-based document type. We believe that an automatic decision model extraction from documents can be initiated from a set of classified documents. Although in this study we are not proposing the whole research methodology to extract decision models, we propose the infrastructure for the classification of documents including rules. As these documents provide the dos and don'ts in everyday work, they are widely known as *Code of Conduct*. These regulations are use cases of how decision making takes place in several managerial levels. Two examples of these documents are presented in appendix B of this thesis. Since two of the main constructive elements of a decision model are the decision input and the decision task, categorizing and understanding the structure of different input and decision task types is vital in this type of research. This research is oriented about the Code of Conduct, in order to provide a clear overview of different rule types and action-prerequisites, which can potentially detect distinctive rule classes for incoming documents.

Similar to all text analysis applications, the first step is determining how to introduce the text data to the machine. In contrast to complete manual approaches in studying the structure of a Code of Conduct, we have identified one which has categorized the document using four pre-assembled groups [99]. That research has focused on a manual decision modeling approach that extracts the input to a decision task and later on, understand the decision logic using mathematical equations. We have used the definitions of categories from this research and used as the basis of a novel automatic machine learning approach explained in this study.

Later, we have discussed the application of Vector Space Model. While most studies create a space out of these term-weight vectors, we have proposed a novel feature selection method based on a combination of words and their respective Part-of-Speech tags. Study has shown that supervised text classification is highly context oriented [100] and our findings certify this phenomenon. With current data collected in this study, three types of features from terms (words), part-of-speech tags and terms plus their part-of-speech tags are examined. We believe that our framework for feature extraction using dual tuples of words and relative POS tags provides a leveraged insight on classification of rule-based documents. Respective accuracy results show that our proposed framework is highly accurate in classifying rules into their respective pre-defined category.

The rest of this chapter is organized as follows. Section 6.2 briefly reviews recent research on supervised text classification and legal text analysis. In section 6.3 the foundations of our research methodology is explicitly discussed and model generation is explained. Section 6.4 presents the data structure and annotation along with text analysis tools. In Section 6.5 we have presented the results of our feature selection framework using classification algorithms. Finally in section 6.6 we express conclusions of this study and obstacles we faced, along with the use cases of our text classification framework for further studies.

6.2 Related work

Compliance regulatory checks is widely known as a function of the legal department of an organization. A legal department representative is responsible to inspect every document with legal burden, extract articles that are vital with respect to business goals, and implement a decision making approach using relevant regulations and finally decide whether the document complies with business law and regulations. Since legal department functionality has not attracted much business intelligence research recently, the spread of text analytics applications is not sensible compared to financial, R&D or Marketing activities. In [101], the authors have examined the use of linguistic information on legal document classification and also proved the application of Support Vector Machines in this field. They have also mentioned the imbalanced nature of the dataset. This issue is also affecting the results in this study which will be discussed in details later.

The use of Support Vector Machines (SVM) has proved to be effective in text classification. In case of big number of classes though, or input data with sparse features, the use of SVM would be highly effective with a linear kernel [102]. A kernel is basically the distance measure between feature vectors. Ratner [103], has shown this fact on a classification problem on complicated legal documents, and found that using SVM with RBF (Radial Basis Function)-or more known as the Gaussian-kernel, which is an exponential function of the normal distance measure between two vectors, returns inaccurate results, with a test accuracy rate of only 25%. Using context-free grammars and Natural Language Processing (NLP) in [104], the researchers have provided informative structure features of law cases like participant roles.

Other supervised learning approaches have been conducted in legal document classification as well. Decision tree as an example, has been discussed in [105], where the authors have conducted an experiment with a combination of four classifiers to categorize legal judgments. Their findings show that C 4.5 has reached the best model accuracy, outperforming Naive Bayes and Winnow classifiers.

The main challenge in text classification is mostly focused on sparseness of feature vectors. In other words, since vector space model requires transformation of terms into vectors, a Bag of Words approach is mainly used in feature extraction. While BoW is effectively used in topic selection, legal documents are of various classes. Sparse matrices of BoW are not capable of determining weights of transformed terms. As a solution, many have employed Tf-Idf which is the weighting factor for how important a term is compared as index in the document it belongs to, as stated before [106] [63].

6.3 Research methodology

Having an automatic rule classification systen from documents has been prioritized an applicable challenge by many specialists. The reason lies in the fact that focusing on terms for the classification task and using relative naive feature selection methodologies such as BoW and Tf-Idf would simply be able to capture a limited number of re-occurring document patterns. It is in our interest to address this challenge in this chapter. We propose that the learning compartment of a document classification schema including rules can capture not only the importance of re-occurring rare but significant terms, but also the grammatical structure of these documents expressed in the form of a set of re-occurring POS tags. To clarify the proposition, a study over the document structure in a typical code of conduct is vital. Furthermore, we present the methodology on pattern capturing.

The automatic sentence classification framework we propose requires document category definitions as the infrastructure for class recognition. Hence, we represent the sentence types initially. Manual trials in annotating compliance regulatory checks, as a well-known portion of legal documents, is the legacy approach in document classification. Research platforms like GATE include annotation capabilities regarding this issue [107]. Commercial software *ATLAS.ti* is another qualitative research and analytics tool which provides text annotation and referencing tools [108]. A couple of previous researchers have gradually achieved a categorical representation of rules in business which we are using in this study. While sentence classes in real legal cases might be more complicated and the number of them would be more as well, four general categories of documents are determined in this study. These categories are respectively called *Condition*, *Obligation*, *Permission* and *Prohibition.* The automatic sentence recognition we have implemented in this research requires a clear definition of each type.

A Condition type explains the prerequisite to an action. Conditions are logical expressions which in most cases can be represented with mathematical equations. For instance, a condition to an action in legal decision making might be:

"If employees become aware of any evidence of fraud and dishonesty, they should immediately advise their supervisor"

In this statement, *If employees become aware of any evidence of fraud and dishonesty*, frames a condition to a decision. Conditions might also be formed using *While* and *When*, which express the existence of a constraint. The second part of the sentence represents the Obligation type. Obligations provide the action that is mandatory to be done and in some cases they are accompanied by time constraints. Example of an obligation might be:

"Company information should be used only for company purposes"

Obligations mostly include language *modal verbs. Must* and *Should* are the most used modal verbs in this matter. Note that in many cases, Obligations are in the form of *self-instructive* sentences, in which the ruler, expresses the task to be done in the form of a predicative statement. For instance:

"We believe in data security and integrity as a general principle"

Permissions are the allowance of a task when facing certain requirements. While most of permissions are expressed using *May* as the modal verb, there are other types of permissions. Take into account the following example:

"It is possible to undertake additional surveys in case of double witnesses compared to the initial inspection stage"

Finally, a Prohibition rejects undertaking a task according to some conditions. Most of prohibition samples include *Must not* or *Should not*, while there might be some uncommon forms, similar to permission sentence type. These irregular types follow the self-instructive sentence type. Two examples of prohibitions are followed:

"Request for an appeal
before the constitutional decision of the court is forbidden"

,

"Employees must not accept any type of gifts from company guests"

Although the grouping looks straightforward, conflicts might arise while choosing the real category for a sentence. This means that a sentence might have characteristics of two or more groups. Some ambiguous types of documents in this regard are discussed in the discussion section.

To answer the research challenge in capturing the maximum level of rule-based document patterns for an efficient classification schema, we propose that determining the structure of a rule provides insight on automatic categorizing capabilities of a text classifier. A classifier which is being trained by various forms of rule structures, can further differentiate between the pre-requirements of a rule-rule criteria and the context of the rule itself. This means, our application field enhances the potential of an automatic decision model extraction from text. The resulting platform, would enable legal departments to search relevant decision making approaches, identify a matching case, conform respective criteria with current information at hand and finally decide on undertaking a task or taking a path as the decision model suggests.

To sum up, we are implementing the classification framework on codes of conduct of multiple business types. In this study, the rule classification begins with a preferred code of conduct as the input document. Data as the training input set, has been annotated precisely, considering grammatical and semantic word based inspection. This means, during the annotation phase we have taken into account, multiple roles a certain expression and word might take, reducing the type assignment error to the maximum. The annotation is performed by the student in this study. Using the context of different structures presented above-concerning dos and donts-the annotation was performed. Each sub-part of a sentence which has certain criteria of a sentence type was saved with their relative class in a csv file to form the training dataset. Sequentially, the back-end technology for analysis is implemented in Python, employing NLP techniques embedded in the Python packages, called NLTK [109] and Scikit-learn [110] (sklearn). Pre-processing enablement was made possible by NLTK and learning procedure for model selection was conducted using scikit-learn. Our novel feature selection methodology is based on a combination of the term and their relevant part of speech tags. Each feature type is used to form the Tf-Idf matrix which will be used for training models using different classification algorithms. Furthermore, the models generated would be tested to see how accurate the performance is. Additionally, we employed 10 fold cross validation to validate our results. In the next section, fundamental knowledge in perceiving the novel feature selection methodology in this study is presented.

6.3.1 Feature selection/extraction fundamentals: hidden patterns

Bag-of-words, as the simple representation of terms, without specific grammar settings in a document is the foundational feature extraction methodology in text mining as we know. One approach in transforming the terms in BoW is to deterimine that existence or absence of a word is translated into 1 or 0, which later forms a vector for each document with 1 and 0. Additionally, term frequency, or the number of times a term is occurring in a document or a set of documents can represent the text in a vector or a matrix of vectors. In other words, we can think of a bag in which all the words of the whole document exist. This bag acts as the matrix, which includes several vectors, representing each sub document, which is in most cases, a sentence. Harris [65] introduced bag of words models for language model representations. Later on, Naive Bayes proved to be effective in model generation, with a bag of words feature extraction methodology [111]. The lack of performance capabilities arise from the fact that bag of words assumes a network of words, which are meaningful to a certain subject. Spam email detection is based on such a theory. Consider an article reporting about a storm. This article would probably contain words such as *wind*, *weather*, *temperature*, *danger*, etc. Bag of words is sensible to the reoccurring words in specific domains and is able to detect similar documents containing words from a network. On the other hand, when the number of terms-or features-is big or in cases where terms do not necessarily represent the identity of a document, bag of words would return deficit predicted categories. In this case, since many of words inside the bag of category do not exist in the new document necessarily, the feature matrix would grow to sparsity, and it contains a lot of zeros. Sparse matrices will not produce efficient models for category prediction. Dimensionality reduction is the practice which would lower the feature subset, in order to decrease the training time and optimize accuracy [112]. A code of conduct contains a large variety of words, which are specific to a certain business type. Hence, bag of words would include a great number of words, and feature matrices with huge dimensions. Here, we represent the combination of words paired with their respective part-of-speech tags and Tf-Idf as the feature extraction method. The use of Tf-Idf is justified based on the explanation presented in section 4.3.

To review the concept of part-of-speech tagging, Stanford parser [113] is currently the research standard platform in assigning certain tags to the terms. An English POS parser, is trained using a corpus-a set of documents-of English documents, and later on is able to tag further input documents. Two of the state-of-the-art English parsers, namely Stanford and Berkley, produce results which can be tested using Penn Treebank [18] [114]. A standard English parser output regarding a regulation follows:

"In every instance where improper behavior is found to have occurred, the company will take appropriate action"

Parsing:

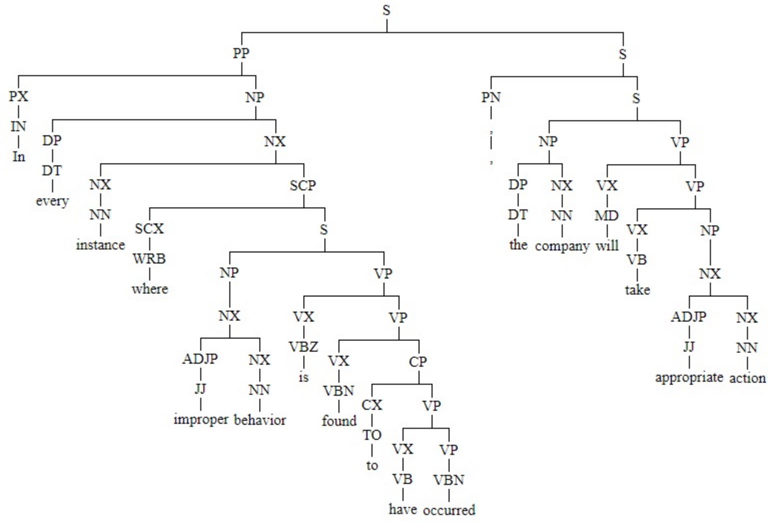


Figure 6.2: Parse tree for the regulation

Tagging:

In/IN every/DT instance/NN where/WRB improper/JJ behavior/NN is/VBZ found/VBN to/TO have/VB occurred/VBN ,/, the/DT company/NN will/MD take/VB appropriate/JJ action/NN

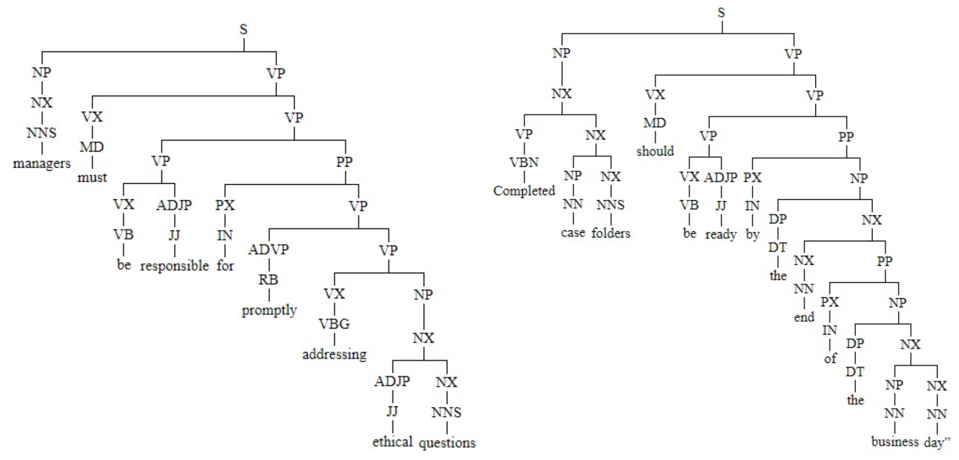
According to our proposition stated before, the grammatical structure of each type of regulations, discussed in previous section, remains the same, regardless of the certain terms occurring in each. Respectively, POS tags would remain intact, resulting in homogeneous parse trees. Consider the following two regulations:

Obligation 1: ”Managers must be responsible for promptly addressing ethical questions”

Tagging:

Managers/NNS must/MD be/VB responsible/JJ for/IN promptly/RB addressing/VBG ethical/JJ questions/NNS

Obligation 2: ”Completed case folders should be ready by the end of the business day”



(a) (b)

Figure 6.3: Parse trees for Obligation 1(a) and 2(b)

Tagging:

Completed/JJ case/NN folders/NNS should/MD be/VB ready/JJ by/IN the/DT end/NN of/IN the/DT business/NN day/NN

As we can see, the central core of both regulations, "NNS MD VB JJ", is similar. Also, both grammar models include words before, in between and at the end of their core. Since these longer sequences of POS tags provide further information on the context of assigned tags, we will expand the Tf-Idf weighting measure of N-grams for POS tags. This means, we believe generating N-grams of POS tags would capture the maximum number of similar patterns for each document type. In the next section, we will discuss the use of N-grams in this application.

6.3.2 N-grams out of POS tags: the novel approach

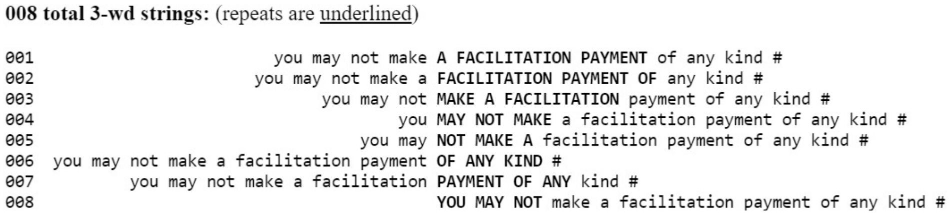
Composing N-grams has recently experienced a major breakthrough in language processing. Assuming constant structures for certain rule types, as previously stated, leads us toward building an expanded knowledge extraction system using lexical N-grams which are generated from POS tags. According to our assumption earlier, we have trained the classification model based on these N-grams. We also tested two other scenarios with using only words, and finally the combination of words plus relative POS tags for feature extraction. Strong relationship has been proved to exist between N-grams created from POS tags and the genre of the text [115]. Our experiment focuses on N-gram generation in a range of

$$N = [2, 5]$$

Since the smallest POS tag in Penn treebank is composed of two characters, the lowest value is 2. We have defined the high value of five since we believe amounts bigger than 5 would tend to form sequences of entities that are more like sentences or expressions rather than words. Consider the following statement:

"You may not make a facilitation payment of any kind"

We have defined and annotated this example as a *Prohibition* rule. Composing word N-gram features can lead to following tri-grams in figure 6.4, being part of the N-gram range for $N = 3$:

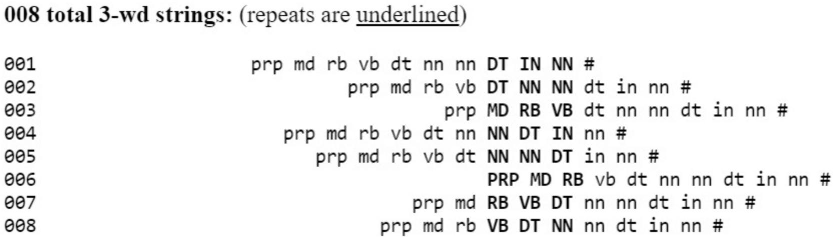


008 total 3-wd strings: (repeats are underlined)

```
001                        you may not make A FACILITATION PAYMENT of any kind #
002                      you may not make a FACILITATION PAYMENT OF any kind #
003                             you may not MAKE A FACILITATION payment of any kind #
004                                     you MAY NOT MAKE a facilitation payment of any kind #
005                                 you may NOT MAKE A facilitation payment of any kind #
006  you may not make a facilitation payment OF ANY KIND #
007          you may not make a facilitation PAYMENT OF ANY kind #
008                                          YOU MAY NOT make a facilitation payment of any kind #
```

Figure 6.4: Word tri-grams generated from the prohibition rule

N-grams of POS tags can be generated from utilization of sklearn Python package. In this process, text data will be tokenized-split to words-and then the POS tagger included in the package will tag words. Furthermore, the tagged set will be used as the N-gram generator. We can see an example of the POS N-grams for $N = 3$ generated from the aforementioned rule in figure 6.5:

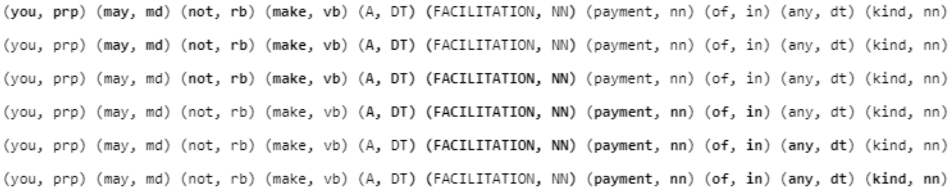


008 total 3-wd strings: (repeats are underlined)

```
001              prp md rb vb dt nn nn DT IN NN #
002                       prp md rb vb DT NN NN dt in nn #
003                                prp MD RB VB dt nn nn dt in nn #
004                 prp md rb vb dt nn NN DT IN nn #
005                    prp md rb vb dt NN NN DT in nn #
006                                    PRP MD RB vb dt nn nn dt in nn #
007                             prp md RB VB DT nn nn dt in nn #
008                          prp md rb VB DT NN nn dt in nn #
```

Figure 6.5: POS tri-grams generated from the prohibition rule

We have customized the N-gram generator to determine a dual feature perspective, to combine the words and corresponding POS tags to construct a third set of features. As a result, an example of N-grams of dual tuples of word and POS tags for $N = 4$ can be visualized in figure 6.6. In all N-gram features, N-grams are visualized in bold format.



(you, prp) (may, md) (not, rb) (make, vb) (A, DT) (FACILITATION, NN) (payment, nn) (of, in) (any, dt) (kind, nn)

(you, prp) **(may, md) (not, rb) (make, vb) (A, DT)** (FACILITATION, NN) (payment, nn) (of, in) (any, dt) (kind, nn)

(you, prp) (may, md) **(not, rb) (make, vb) (A, DT) (FACILITATION, NN)** (payment, nn) (of, in) (any, dt) (kind, nn)

(you, prp) (may, md) (not, rb) (make, vb) **(A, DT) (FACILITATION, NN) (payment, nn) (of, in)** (any, dt) (kind, nn)

(you, prp) (may, md) (not, rb) (make, vb) (A, DT) **(FACILITATION, NN) (payment, nn) (of, in) (any, dt)** (kind, nn)

(you, prp) (may, md) (not, rb) (make, vb) (A, DT) (FACILITATION, NN) **(payment, nn) (of, in) (any, dt) (kind, nn)**

Figure 6.6: Dual quad-grams of words and POS tags from the prohibition rule

At the end of this phase, a converting function creates a Tf-Idf matrix from N-grams. Output features are input to a set of classification algorithms.

6.4 Data structure and Analysis

We gathered multiple codes of conduct from various industries. Text format extracted varies from txt and pdf to doc and docx. Since uni-format text analysis is preferably desired, we converted all documents into pdf format. The corpus at hand, includes text, figures, headings and references. In this research though, we are only interested in text. As a result, we implemented a Python snippet which extracts meaningful text only from a pdf to be classified later.

A Code of Conduct consists of fields of concern in business blueprint, and provides guideline in each case, ensuring a safe work environment and business security. Each section has headings, and a set of regulations. This procedure will be executed for each new input document which should be classified. The training data samples are manually annotated by the student in this study. These instances receive one of the 4 pre-defined classes based on the context of the text and the sentence structure. Manually annotated and classified training data, extracted from Codes of Conduct of different industries in csv format, is fed into the machine to train models. Each document from the training corpus is read, tokenized and tagged using NLTK. This Python package, implements these preprocessing steps using classes. As an example the nltk.tonenize package manages the class nltk.tokenize.api.StringTokenizer as part of its tokenizing capabilities. Due to analysis requirements, the machine then creates N-grams from words, POS tags and the combination of both. Table 6.1 provides a short sample of the training input data.

Statement	Rule Type
If you believe that your own or another employee's behavior contravenes the values and principles of conduct outlined in this Code	condition
Keep the private use of the Company's IT infrastructure, including e-mail, Internet access, and telephones within appropriate limits	obligation
We do not make any financial donations to political parties or similar institutions, or to individuals	prohibition
You can, however, take reasonable time off without pay for such activities if your IBM duties permit and it is approved by your manager	permission
Information on how to report and protect intellectual property can be found at the Intellectual Property & Licensing site	permission

Table 6.1: A short sample of training data

Tf-Idf vectorizer, is the converting function which transforms raw documents to a feature matrix. Tf-Idf is used in this research, to find out determinant features in all three feature-type sets we have defined. *TfIdf Vectorizer* function is employed to determine such an effect.

In this phase, it is necessary to understand how the *TfIdf Vectorizer* function creates the input for the classifiers. First, for the N-gram range [2,5], a certain class method *fit_transform* from sklearn *TfidfVectorizer* learns vocabulary, creates term-document matrix and calculates TF and IDF and returns this matrix. This step is implemented on the training dataset. This is part of pre-learning procedure and elements of the matrix are three types of features we identified to train the classifiers. This means that we have experimented the effect of three feature types separately each time on learning approach and trained the classifiers accordingly. Furthermore the method *transform* executes the converting function and transforms all documents from the test dataset into document-term matrix, so they can be properly prepared as the classifier input. We do not limit the number of features in this research. As a result, a document my contain any number of features.

Furthermore, two Support Vector Machine classifiers start the learning procedure. Theoretically, SVM considers the number of features for classification and tries to draw hypothetical lines which keep the samples in each category as far as possible. In other words, for a two feature space, SVM is the line which separates the two categories and has farthest distance from both closest points of two groups. Further test samples are classified based on the place they reside next to the line. For more than two features, hypothetical planes classify the data instead of the line. We chose linear SVM and a non-linear SVM with *gamma* and C parameters of 2 and 1, respectively. We also determined a linear SVM with an error parameter C of 0.025. In order to understand C, recognition of the difference between linear and non-linear SVM is vital. In many cases, with semi-full visual distinction of classes, the SVM classifier line is able to distinguish classes with low error rate. Error reducing efforts lead to non-linear distinctive classifier which curves around a couple of samples to limit the error rate. Figure 6.7 and 6.8 present examples of such a modification.

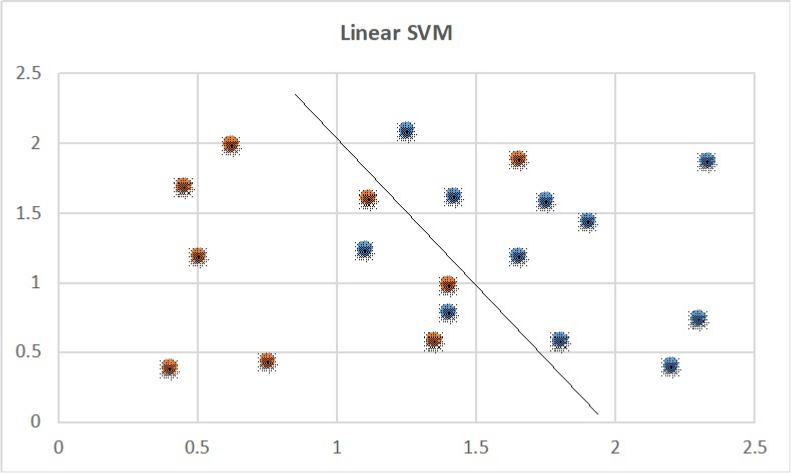


Figure 6.7: Scatter plot of the estimated classifier with optimal distance from closest samples(support vectors) for dual class

A trained classifier as in figure 6.8 is optimistic, since applying it to the test set will not probably return a generalized output. C parameter in sklearn, is the compromise between the classification for training set samples and the optimal length of support vectors. Gamma parameter, determines the effect of a certain training sample. Lower values of Gamma magnifies the influence of farther samples from the classifier, while higher values of Gamma concentrates on the closer samples. As a result, defining a high value for Gamma leads to a more curved classifier, which is a customization for specific data types. It is necessary to note, that data is gathered cumulatively, in multiple phases.

The other algorithms implemented include AdaBoost, Random Forest, Multilayer Perceptron and a set of Naive Bayes family algorithms. The AdaBoost belongs to Boosting family algorithms, which as discussed earlier, aggregate a group of weak learning algorithms into one powerful learner. Simply stated, AdaBoost performs a set of iterations on the training set data samples using a base classifier, calculate the training error, compute and apply a specific update in the distribution and finally reiterate through other samples using an updated learner

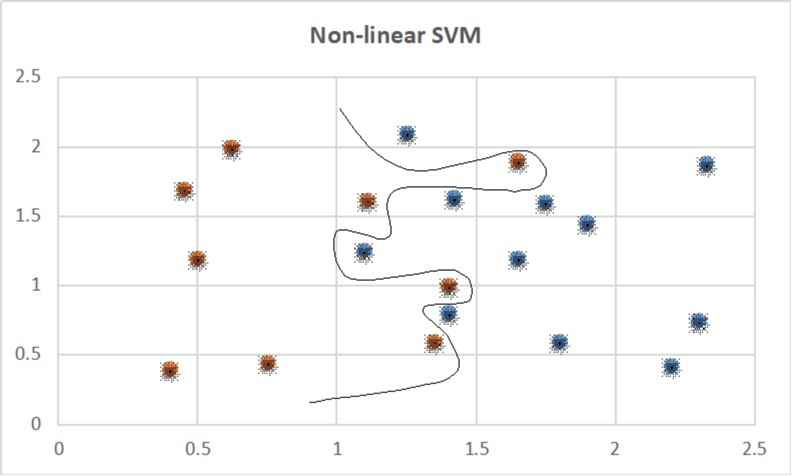


Figure 6.8: Scatter plot for dual class with non-linear SVM classifier

[116].

Random Forest classifier which is built on the ground structure of Decision Trees, gathers knowledge using a number of them to do the prediction task using the vote of the majority of decision trees. In other words, each decision tree in our case study gathers historical evidence using the training dataset in order to predict a class for an instance. The final decision would be the class predicted by most of the decision trees. Obviously, considerable results would be achieved by the cost of longer time required for the algorithm to find the most voted class.

The Naive Bayes family classifier in this study include Gaussian Naive Bayes, Bernoulli Naive Bayes and Multinomial Naive Bayes. Based on the Bayes rule, these classifiers learns how the class attribute based on an assumption that all features defining a class are independent from each other. For instance,in document classification certain terms would represent the class such *social*. Naive Bayes assumes this terms are all independent of each other, contributing in class definition. Yet, many of the terms may belong to other classes as well. This deficiency originates from the naive assumption of Bayes Theorem. However, with lower number of classes, Naive Bayes still performs considerably. An example is the application of Naive Bayes in classification of statements to three classes of positive, negative or neutral.

Multilayer perceptron classifier is introduced with more details in the results and discussion sections. In the next section, we have illustrated our framework results.

6.5 Results

In this study, we trained a couple of models, providing them with features from N-grams of words and relative POS tags. The classification algorithms we have tested are SVC (implementation of SVM in sklearn) family, Random forest, Adaboost classifier, Multilayer perceptron and a group of Naive Bayes classifiers. We employed 10 fold cross validation to validate our test set results. This functionality is represented in the code.

Information on the parameters used for each tuning the classification algorithms is necessary for reproducibility of the results. Parameter representation of all algorithms are based on *sklearn* Python package documentation[1]. For the linear SVC algorithm, we used the default parameters. Specifically, the amount assigned to *kernel* is *linear*. The second SVC linear algorithm was implemented with the default parameters as well. The only exception was the c parameter, which gets the amount of 1.0 by default. But in this experiment the amount is assigned the amount of 0.025. Cross-validation enhances the amount of C and error parameter with for different amounts of gamma parameter in the non-linear SVC algorithm. For the RBF kernel used in this algorithm as discussed earlier, we used the amount of 2 for Gamma parameter. For the AdaBoost, Random Forest, Multilayer perceptron and Naive Bayes family classifiers we used the default parameters which are available to inspect in the Python package *sklearn* documentation reference.

Among SVC family classifiers, linear SVC, which is the major focus point of this chapter, returned considerably high accuracy. Using N-grams of words, SVC reached the average score of 72%, compared to 58% and 60% accuracy levels, for linear SVC with C parameter of 0.025. As it is explained in the previous section, we can see linearity of the classifier has changed, which means it is more biased to find the right class of the training samples, while the accuracy decreased compared to a total linear SVC, since the classifier was not able to classify the test set with the same performance. Due to the fact that text data is dividable linearly in most cases, a linear kernel for SVM proves to be more effective. Also, we need to consider the fact that the number of features in this study dataset is relatively high. Hence, mapping data to higher dimension space will not necessarily return better results [102]. Non-linear SVM in this case, would require more resources only. A Multilayer perceptron classifier as well, shows great results. Being a supervised Neural Network, it requires more time for learning procedure, but it is highly efficient to work with many features, as it is in this case study, as it is traditionally proved, as in [26]. Table 6.2 presents the accuracy of classifiers tested in this case.

Selecting N-grams of POS tags, has shown to slightly improve the Non-linear SVC accuracy, while the accuracy of Linear SVC remains intact. All the other models show lower accuracy compared to features of word N-grams. Random forest classifier is not competitive in performance, while dealing with text data, due to the sparsity of data. Such an issue, causes weak sub-trees to be shaped, result-

[1] http://scikit-learn.org/stable/

Classifier	Average Accuracy
SVC Linear	72.28%
SVC Linear C=0.025	58.50%
SVC gamma=2 , C=1	60.07%
Ada Boost	75.24%
Random forest	70.62%
Multilayer perceptron	79.35%

Table 6.2: Accuracy of classifiers for N-grams of words as features

ing in a deficit performance of the whole algorithm. Examples exist in previous research on this topic [117]. The results are presented in table 6.3.

Classifier	Average Accuracy
SVC Linear	72.28%
SVC Linear C=0.025	58.50%
SVC gamma=2 , C=1	60.58%
Ada Boost	64.19%
Random forest	67.31%
Multilayer perceptron	72.98%

Table 6.3: Accuracy of classifiers for N-grams of POS tags as features

The combination of words and their relative POS tags, has greatly influenced the learning capabilities of our classifiers. In our experiment, we fed the N-grams of tuple features, to the linear SVC classifier. As a result, we witnessed a major improvement of 12% in model accuracy. All Ada Boost, Random forest and Multilayer perceptron classifiers as well are improved in classification performance compared to N-grams of POS tag features. Their respective accuracy has improved by 8%, 12% and 13%. However, accuracy is declined for Ada Boost classifier by 3% compared to using N-grams of words as features, while Random forest and Multilayer perceptron show an improvement of 9% and 6%, respectively. Generally, using the N-grams of dual tuple features, as it is in our third case, has returned the best performance in legal document classification although we see some irregularities in the performance prediction for Random forest. Table 6.4 presents the results of the experiment for accuracy of classifier using N-grams of tuples from word-POS tag.

As we can see, the Multilayer perceptron classifier has reached the highest accuracy in all three experiments. Defined as a supervised Neural Network, Multilayer perceptron is able to determine significant number of features as input neurons.

Classifier	Average Accuracy
SVC Linear	84.48%
SVC Linear C=0.025	58.50%
SVC gamma=2 , C=1	60.50%
Ada Boost	72.54%
Random forest	79.35%
Multilayer perceptron	85.18%

Table 6.4: Accuracy of classifiers for N-grams of tuples from word-POS tags

Time required to handle these features is relatively higher in this case. Still, introduction of this classifier in this research is only for comparison purposes and the study of neural networks is out of the scope of this research. In neural networks, one rather important matter is the calculation of the gradient which defines the weights to be used in the network for each input. Since our training set is rather small in size of samples, we have used an *output weight optimizer*, which reduces the time for training and validation and is an alternative to the *backpropagation*. This optimizer is defined using parameter *solver* set to amount *lbfgs*. According to scikit-learn package preferences, *lbfgs* is the optimizer from the quasi-Newton methods, which aims to find the minimums/maximums or zeros of a non-linear function. Scikit-learn also states that for datasets including thousands of samples for each classes are optimized better using the default amount, the stochastic gradient-based optimizer *adam* [118] as the *solver* in terms of training time and validation accuracy. Yet, for relatively smaller datasets, such as the one we are using, the degree of convergence of the neural network would be optimized using *lbfgs*. The other possible amount other than the two mentioned is the basic stochastic gradient-based optimizer *sgd*. Figures 6.9, 6.10 and 6.11 represent the visualization for the distribution of 10 fold cross validated results.

We have gradually extended the dataset size used in the three experiments. The number of samples in our dataset increased from 300-presented in appendix D-to 1200. As a result we were able to observe the influence of dataset size on the experiment. A slight decline in our accuracy results occurred, given the fact that the final dataset is four times bigger than the initial dataset. The reason lies in the fact that the initial dataset contained very few samples from a specific class. This uneven distribution of the samples among classes provides certain classes with an artificial benefit of affecting the classifier. The proportion of certain classes, for instance the Prohibition class to the whole dataset is not constant in the initial and final dataset. Hence, the initial dataset accuracy is rather unreal and mainly framed by the low number of instances in certain group and high number in others, while with a more even distribution of samples among classes in the final dataset, a reasonable number of classified samples is assigned to each class, which somehow normalizes the degree of significance of each class for the classifier. Figure 6.12, provides a more detailed overview regarding the initial and final dataset classification results.

As explained, using stratified K-fold cross validation in data mining problems ensures a relative balanced selection of samples from all available classes. Yet, imbalanced training datasets in terms of unequal number of samples from each class can still impact the selection procedure. In other words, since in our dataset the number of records for each class is not the same, 10 fold cross validation still keeps the relative proportion of sample size in data selection. As a result there will be folds of data in which there are less number of records of a specific class compared to the others. The opposite case is also probable, where the number of samples from a certain category is very high, or they compose the whole class. Such a case will result in over-fitting of the model towards specific classes. We additionally validated the results using 5 fold cross validation, instead of 10 folds. Considering the fact that lowering the number of folds would somehow increase the number of samples from low impact class, we observed worse results. The outcome supports the assumption of over-fitting models to specific class with high population in a major number of the folds. We have presented the accuracy model of three types of Naive Bayes classifiers, as a basis of comparison for our models. These three include Gaussian, Multinomial and Bernoulli Naive Bayes with respective accuracy of 73%, 61% and 60%. Model accuracy for these algorithms for both initial and final datasets is visualized in figure 6.11 as well.

With respect to 10 rounds of data split for 10-fold cross validation, a confusion matrix for each round is generated for each classification algorithm. For the models trained with most accurate features which is the third type, the confusion matrix of results in the first round, for linear SVC, Adaboost, Random Forest and Multilayer perceptron classifiers are visualized in tables 6.5, 6.6, 6.7 and 6.8. These models are picked based on their accurate performance. For all models and all rounds, confusion matrix of results for each model is provided in appendix E through appendix I. Additionally, in order to assess the statistical significance of the results, two of the most precise algorithms rather than Multilayer perceptron, which are SVM and Random Forest are taken into consideration and a paired samples t-test is performed on accuracy and recall measures gained by these two algorithms when trained using the third type of features. Using the results for 10 rounds, a *two tail p-value* of 0.000164286 is calculated for the t-test of accuracy results. Since this amount is smaller than 0.05, the difference between SVM and Random Forest is *statistically highly significant*, as it is as well less than 0.001. For the recall measure, this amount is 0.025019297, which is again smaller than 5% and it confirms the validity of the advantage of SVM over Random Forest. The results for 10 rounds of the analysis used in calculation of the paired samples t-test are presented in tables 6.12 and 6.13.

The output of our classification framework, saved in the form of classified sentences in four categories mentioned earlier, is available in csv format. The whole code, including PDF text extractor, pre-processor, feature extractor using N-grams, model training and the actual text classifier is also available in case of demand. This research output is the properly prepared input for the decision

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**9**	8	0	0	17
	Obligation	2	**55**	1	10	68
	Prohibition	0	10	**10**	0	20
	Permission	0	8	2	**2**	12
	Total	11	73	13	12	117

Table 6.5: SVC Linear 1st round (one data fold of 1200 samples) confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**13**	4	0	0	17
	Obligation	0	**66**	2	0	68
	Prohibition	0	2	**18**	0	20
	Permission	0	10	1	**1**	12
	Total	13	82	21	1	117

Table 6.6: Adaboost classifier 1st round (one data fold of 1200 samples) confusion matrix for 4 classes

model extraction framework from text.

In order to dissolve the complexities in terms of classification output, in the next section we have presented a couple of rule-based phrases and reviewed the error types with examples the classification models have returned.

6.6 Discussion

Like any other text classification model, the models presented in this study face incurred predictive uncertainty in the form of wrongly classified samples of phrases. Looking back at previous section, we can see that the most precise classification models are trained using the dual features of terms and respective POS tag with the model accuracy of 84.48% for the linear support vector machine algorithm which shows pretty close results to Multilayer perceptron with the accuracy of 85.18%. Artificial neural networks are highly accurate and capable in classification of data samples with substantial number of classification features. In case the SVM results were relatively weaker compared to the Multilayer perceptron, we could say the neural network is the best algorithm in the context of this experiment and among the presented algorithms in text classification for rule-based documents. Yet,

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**12**	5	0	0	17
	Obligation	0	**66**	2	0	68
	Prohibition	1	8	**11**	0	20
	Permission	1	7	2	**2**	12
	Total	14	86	15	2	117

Table 6.7: Random Forest classifier 1st round (one data fold of 1200 samples) confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**15**	1	1	0	17
	Obligation	0	**64**	4	0	68
	Prohibition	0	1	**19**	0	20
	Permission	0	8	3	**1**	12
	Total	15	74	27	1	117

Table 6.8: Multilayer perceptron classifier 1st round (one data fold of 1200 samples) confusion matrix for 4 classes

we intend to find the balance between the quality and the time required and as long as the two algorithms showed pretty close results, SVM has showed a better performance while requiring much less time for the analysis.

Inconsistencies in a variety of sentences exist that we need to review and dissolve. In this regards, two issues are to be discovered and discussed. First, complexities occur in the form of ambiguous phrases that inherit the grammar structure from one of the four rule-based statement groups discussed, but actually does not belong to any of the categories. Hence, the phrase categories which are populated with similar grammar structures, lead to trained models that detect these ambiguous sentences as a miss-classified sample and classification errors grows in this way. Some examples clarify the subject in this matter. Check out table 6.9 for a better image.

Document number	Sample document
1	Budget dedication for corporate seminars can be allowed.
2	However, getting the approval from board of directors can take a long time.

Table 6.9: Ambiguity type 1

In table 6.9, document number 1 shows a Permission type sentence in training set of documents. Two root parse tree components of this sentence include the noun phrase *budget dedication for corporate seminars* and the verb phrase *can be allowed*. Since any type of noun phrase at the beginning of sentence can occur with an unlimited number of words, the model is basically trained based on the verb phrase. This means the concentration for learning methodology is on the verb phrase. On the other hand, breaking document number 2 returns a parse structure composed of an initial adverbial phrase *however* and a similar two part noun phrase of *getting the approval from board of directors* and verb phrase of *can take a long time*. Now, since there will be a big number of similar documents in the training dataset with similar POS tag parse structure, the extended majority overcomes the rule category structure and sets itself as the only permission type structure or at least as one of the most used formats. From the human interpreter point of view, document number 1 is clearly a permission. But the second document is of no importance from a rule based decision making aspect. Document number 2 is not a permission but the learning model, having seen a lot of similar POS parse trees, will detect this document as a permission, resulting in an statistical study type I error. While in case of dual features using terms and POS tags, the model enhances learning using terms in addition to various types of POS structure, higher classification accuracy is justifiable compared to POS tags-only features with lower model accuracy of 72.28%. The second type of ambiguity of phrases raises in case of dual or multiple class characteristic inheritance of terms and POS tags. Such a case exits when a part of reoccurring sentence class structure defines a part and another class structure defines another part of the input sentence structure. Consider the following statements in table 6.10.

Document number	Sample document
1	Failing to check if a certain use of company assets is okay is out of the scope.
2	Failing to do so may lead to legal claims.
3	However, if you are also involved in that Google business relationship, it can be very sensitive.

Table 6.10: Ambiguity type 2

The first two of the sample documents here have been wrongly chosen as the Obligation rule type in the output file. The explanation for this wrong classification is directly a matter of mixed structures from two or more classes. Document 1 can be broken into 3 sections independently. A Condition class type of *”if a certain use of company assets is okay”*, another Condition class type of *”failing to check”* and finally the entire body of sentence which is of class type Obligation. The reason for the whole sentence body to be of Obligation type rises from the effective verb phrase of *is out of the scope* and whole set of terms occurring beforehand acts as a coordinated phrase or simply a conjunctive phrase to the verb *failing*. Yet, the human interpreter observes an explicit non-relevant sentence to any of the rule types. From this point of view, document number 1 does not belong to any of the four categories of rule-based documents. Document number 3 is categorized as a Permission class type, simply because the latter sentence component *"it can be very sensitive"* share the structure of a Permission rule type, while it is clear that this document does not belong to any rule-based document classes. The other similar probable complexity is the reversed use of Condition class type, where a default occurrence is determined at the beginning of a phrase, completed by one of the three rule-types. When the Condition type situates after the rule-based types, depending on the class type of the first component, the classification might show irregular behaviors and emit the second part class. Finally, ambiguity rises between two specific class types in some cases: Obligation and Prohibition. Such an ambiguity rises not so unexpectedly since in many cases, the interpreter detects the obligatory of a task that should be or should not be undertaken, while machines totally depend on the interpretation of the user reflected in the training set. Consider the following two examples.

Finally , do not start your own business if it will compete with Google.

and

Ensure Financial Integrity and Responsibility.

At the first glance, the above first sentence is detected to be of class Obligation, as the model has predicted as well. Yet, it also follows the Prohibition class type characteristics, as it forbids a certain task. On the contrary, the second sentence is categorized as a Prohibition, while it is obviously an Obligation rule-type. A relative category detection in this case totally depends on the training set samples and the annotation procedure.

The second issue is the sentences with a wrong class tag with no clear explanation. In a lot of cases, these irregularities happen due to deficit tokenization of the documents or the effect of high sample number of the major class in training dataset. In the first case scenario, parts of sentences are frequently chosen as a class type, while these phrases are not even a full sentence. In the second scenario, when the sample fold in determination includes a high number of sentences from a single category, the model tends to be trained based on that class. Both of these scenarios result in type II errors. Table 6.11 represents some of these examples.

6.7 Conclusion

This study conducted experiments on a novel feature extraction framework, which focuses on the structure based consistency of regulations. We proved that the combination of words and their respective POS tag is an efficient feature extraction method for legal document classification. We observed that the grammatical structure of legal documents expressing regulations, experiences very little change document-wise. As a result, we employed this corpus characteristic, to develop a text classification system, that categorizes the rules into 4 pre-defined groups. Among these, the *Condition* sentence type, acts as a pre-requisite, for the other three groups. In other words, the *Condition* sentence type, provides insight regarding an action which is performed by a role player in business environment. Taking into account the constitutive components of a decision model, which are the decision input, decision logic and the decision task, we have figured the infrastructure platform for detecting two of the decision model components which are the input and the task. We believe that the initial step in an automatic approach towards text unit recognition is providing info on the possible categories the *decision task* might belong to. Further, the task can be connected to a specific decision input. Consequently, relative decision logic coordinated with the task is extracted. Our approach brings the fundamental classes of the three specific decision types, in the form of a code of conduct, and helps finding the decision input, in the form of a condition to the decision task.

We tested three feature selection methods in order to train a couple of models for category detection. Among them, the N-grams made out of dual tuples of words and their relative tags showed the greatest performance, proving the fact that features from both words and POS tags are the most informative while training text classifiers. In other words, our theory of consistent sentence structure provides the added value of knowledge representation, jointly with words composing the expression.

The process of annotating training data required relatively long time-approximately one week-and considerable resources. Although we employed platforms like GATE, human supervising which asks high levels of precision, is time consuming. Additionally, the high number of features in text mining, demands intense processing power. If possible, feature reduction methods which eliminate less informative features are necessary. Although our framework is resource optimized for data samples of thousands and more, which is the case of dealing with a couple of new documents, we might need to employ feature reduction for handling millions of documents, which is normally improbable.

Further development of the decision model extraction based on our text classification framework would be a promising research motivation. Moreover, research over more complicated structures in rules leads toward a broader knowledge perspective, which later returns added value in precise decision model extraction frameworks.

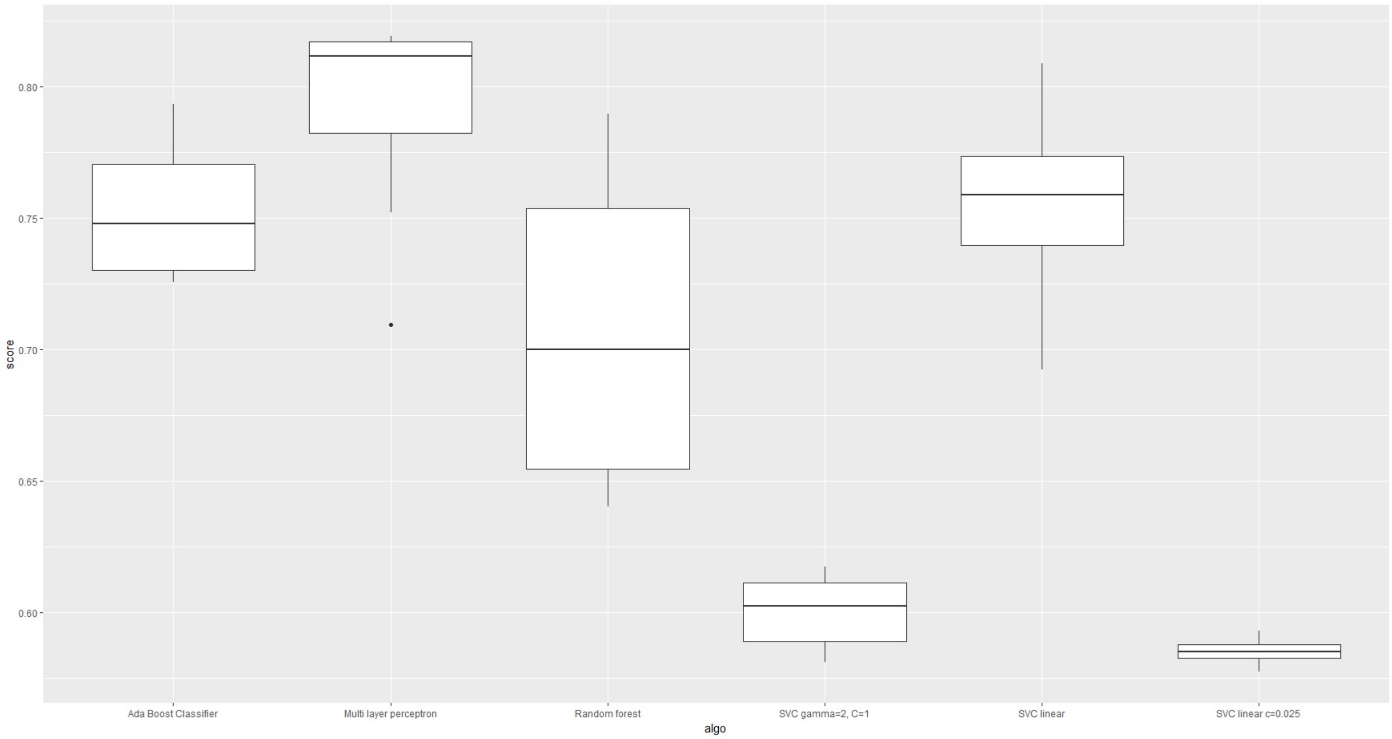


Figure 6.9: Distribution of accuracy results using 10 fold cross validation for word features

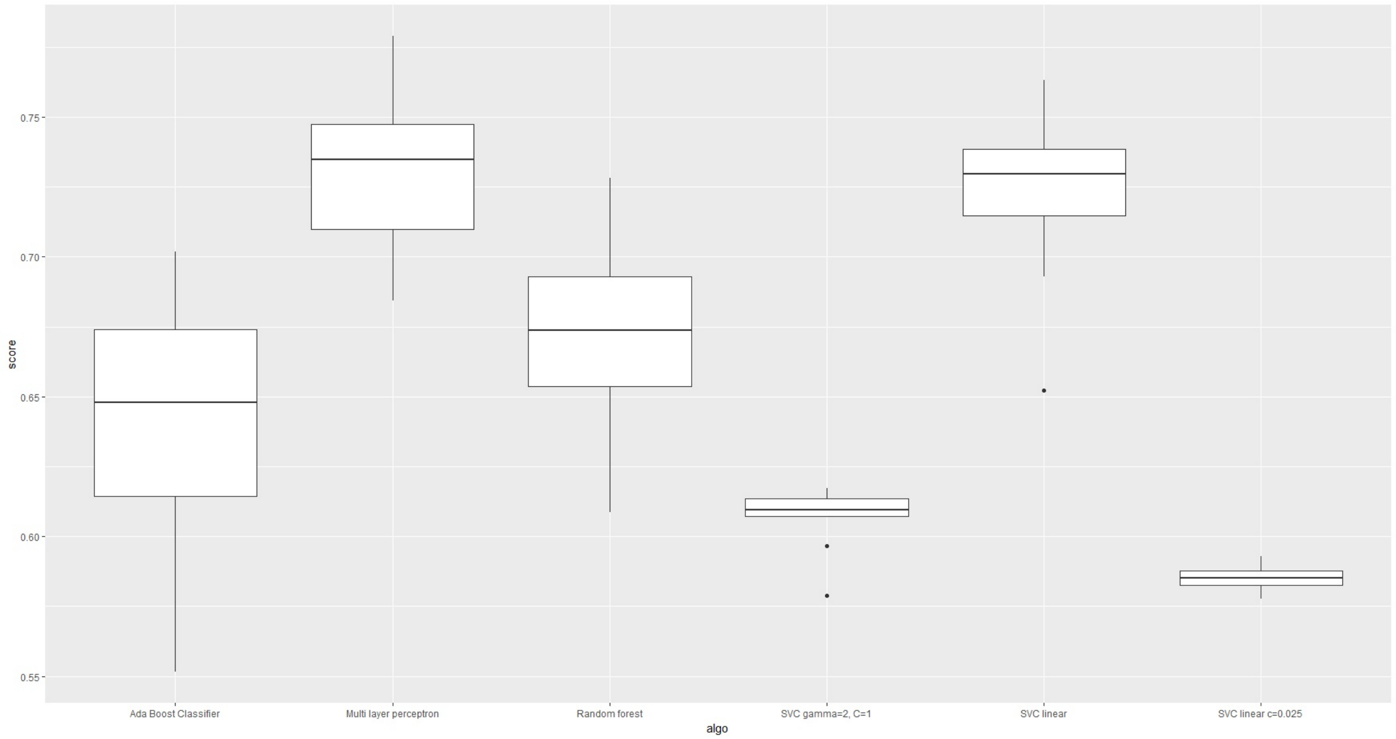


Figure 6.10: Distribution of accuracy results using 10 fold cross validation for POS tag features

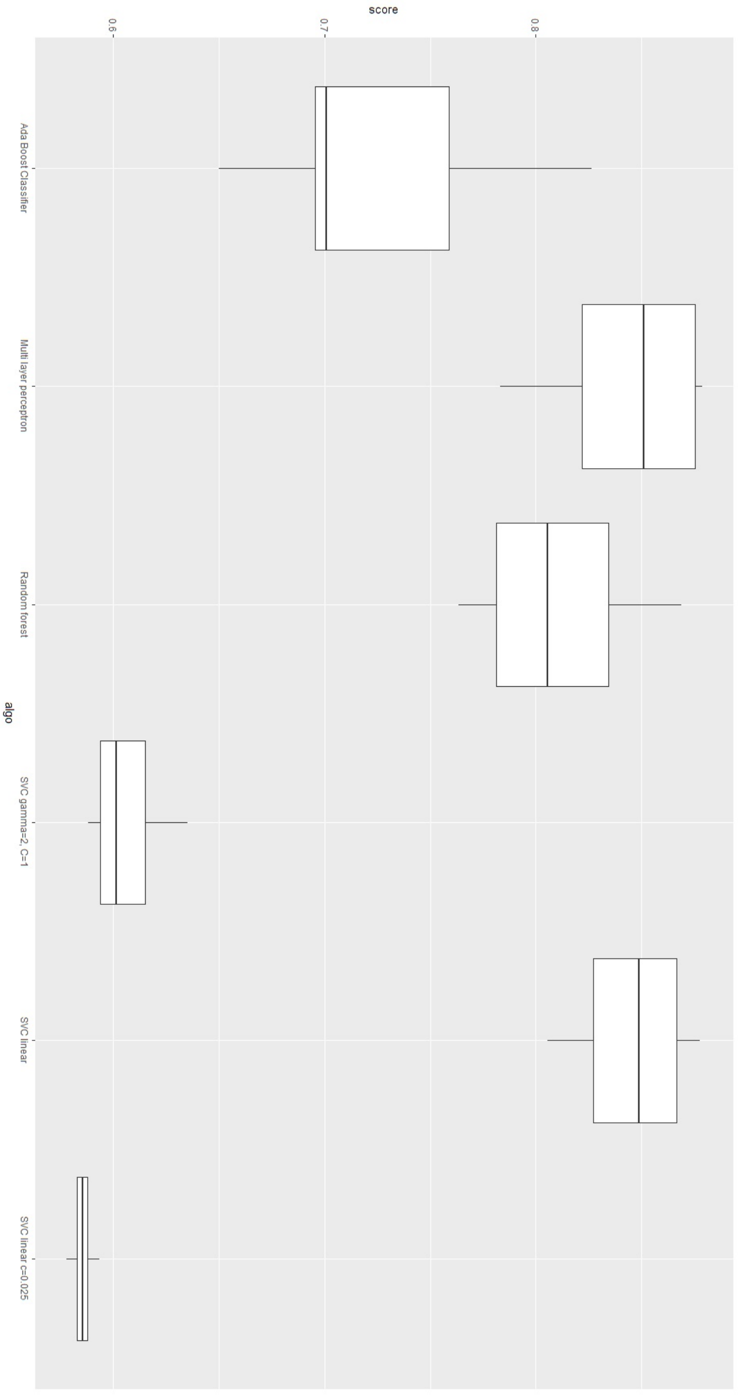


Figure 6.11: Distribution of accuracy results using 10 fold cross validation for tuples of word, POS tag features

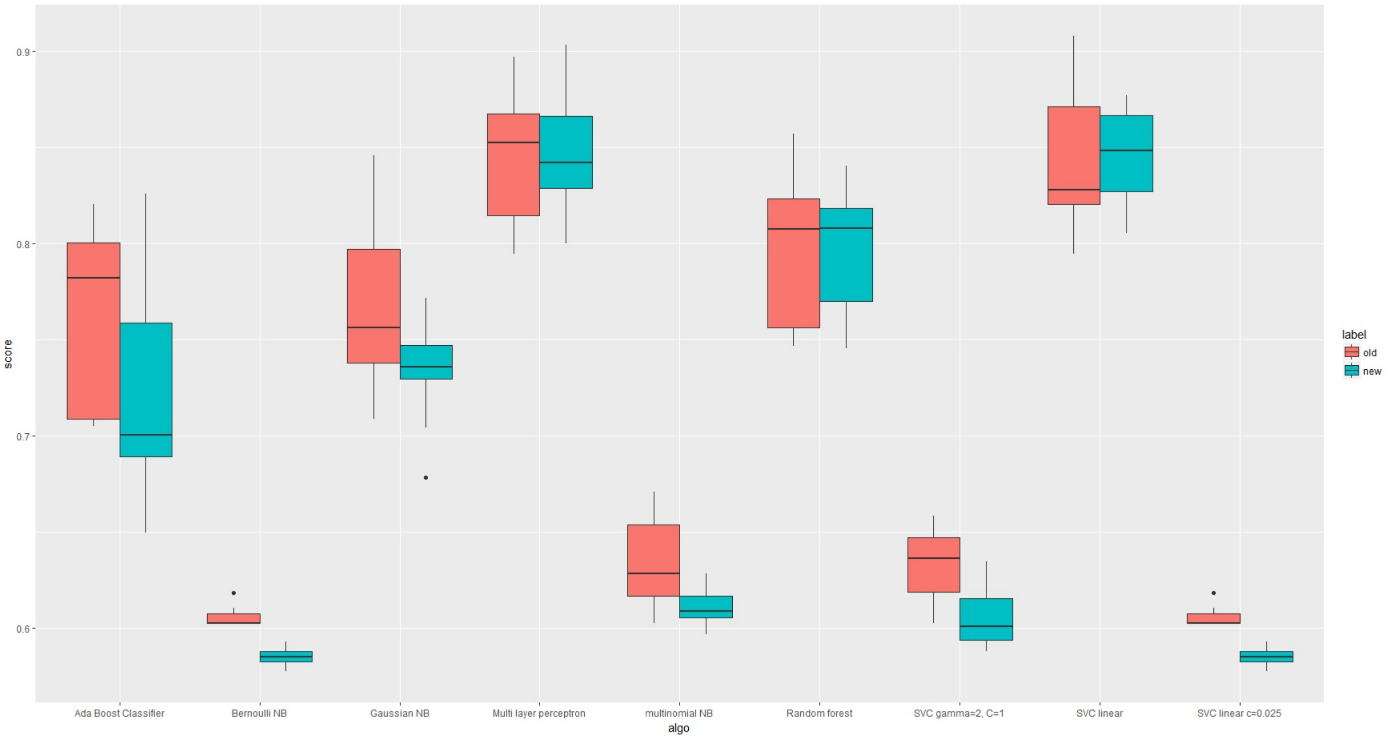


Figure 6.12: Model accuracy results comparing initial (low number of instances) and final (higher number of instances) datasets

Document		
Document number	Content	Proposed class
1	Preserve Confidentiality usually fine.	obligation
2	Confidential Information financial, product and user disclose it outside of Google without authorization.	obligation
3	close any confidential information .	permission
4	Our ability to continue these practices depends on how well we conserve company resources and protect company assets and information.	obligation
5	A word about open source Google is committed to open source software development.	obligation
6	Stay in contact with Ethics & Compliance or Finance if you have any questions.	condition
7	Signing a contract on behalf of Google is a very big deal.	condition

Table 6.11: A sample of miss-classified documents

SVM accuracy	Random Forest accuracy
0.837606838	0.777777778
0.862068966	0.827586207
0.808695652	0.791304348
0.834782609	0.739130435
0.869565217	0.817391304
0.859649123	0.815789474
0.877192982	0.798245614
0.868421053	0.815789474
0.824561404	0.807017544
0.805309735	0.769911504

Table 6.12: SVM classifier accuracy and Random Forest classifier accuracy for 10 rounds used in calculation of t-test

SVM recall	Random Forest recall
0.679656863	0.598284314
0.721023559	0.749915862
0.674537074	0.694893846
0.679879679	0.624962088
0.757644265	0.681185649
0.75414512	0.743652621
0.763012968	0.663159365
0.740428522	0.718183288
0.669995085	0.643536469
0.614390666	0.573795571

Table 6.13: SVM classifier recall and Random Forest classifier recall for 10 rounds used in calculation of t-test

Chapter 7

Conclusion: Remarks and future work

Reaching the final phase of this study, in this chapter we outline achievements of research in the scope of research objectives defined in the introduction section and potential research objectives illuminated by this study findings are elaborated.

7.1 Results

Due to broad change in focus of attention in business intelligence towards facilitated use of data, new data analysis establishments are exploring less considered data types. Pattern recognition in this matter is of substantial interest. Researchers believe data has story to tell, which is not perceived by ordinary means of analysis. Data maintained in the format of text is highly in-use as well in this regard. As a result, more and more text mining applications tend to develop pattern extraction techniques for diversified text formats of any human language and distinct language models. Hence, it is in our research interest to dig deeper into text mining applications that enhance business oriented tasks and procedures. In this field, we defined three main challenges to overcome in order to gain added value in less discovered text mining characteristics.

To overcome the complexities in interpretation of text data generated and dealt with daily in business tasks, a detailed study on nature and role of text data, relevant terminology and specific techniques in handling this type of data is required. We presented the fundamentals of document classification as a widely in-use text mining application and with respect to various approaches and techniques to undertake a document classification task, we introduced the concept of text data quality and the necessity of a transformation function to convert text data into appropriate machine learning inputs. In this regard, the concept of a text classification *feature* was presented and different feature selection and extraction methods were introduced.

To facilitate text classification with massive document sets with many uninformative classification features and to overcome the second challenge, we provided the fundamentals of supervised text classification. Complexity is further increased while data is of two or more languages. To represent a solution, we introduced a commodity classification problem provided by a parcel delivery company in Middle East. The aim was to develop a classification system for a messy database of over a million records with over 50 data attributes based on one of the attributes that is the description of item, containing terms in two languages of English and Arabic. We proposed a series of steps to transform the text into a unified set of language characters. Furthermore, transforming the informative features based on Tf-IDf was performed. In the next step, a couple of text classification algorithms were trained based on an already classified subset of the whole dataset. A set of 70 indicative classes of products-according to client request-were available. We were able to match the records presented in the dataset by a high precision. In this study, the limitations of semi-supervised approaches such as the use of WordNet with classes which are not completely compatible with WordNet word mapping are illustrated as well. We introduced supervised feature selection in text classification and reviewed a couple of supervised feature extraction methods, leading to a new feature selection method represented in chapter 6.

The third and final challenge in this study was to support and provide the initial infrastructure for automated decision model extraction from documents containing decision task, decision input and decision logic. In order to do so, the first step is to have an automatic classification schema for different types of rule-based documents. By determining a solid grammatical structure in these document types, a novel feature selection model based on *N-grams of part-of-speech tags* on these documents is introduced. Combining this feature selection methodology with the terms as features themselves, we introduced as well the main feature selection methodology in this study, and trained a couple of text classification models, including *Support Vector Machines.* The models based on *SVM* were able to classify the rules in a specific rule-based business document, the code of conduct, with an outstanding performance. Classified rules are input to further decision model extraction, as we were able to detect two of the main components of a *Decision Model* which are *Decision task*-as one of the three categories of rules detected-and the *Decision input* as the *Condition type* rule component classified using our classification model.

7.2 Future work

Two main experiments designed and implemented in this study open up new perspectives over the application of data mining techniques in handling text data. Utilization of documents classification in the parcel delivery company case and rule-based documents as fundamentals of decision model extraction, provided clear insight on how text mining can facilitate extracting patterns from daily used text data. In this respect, potential research interests may arise from this study. In document classification for multiple language text for instance, there is place for improvement in classification model accuracy by taking other approaches instead of translation of data into unified models. Extension of NLP techniques to every available language and assurance in receiving comparable results when applying certain NLP techniques to different languages can potentially result in more accurate data inputs. This means, instead of translating all data instances into English for instance, each set of samples from a certain language can be dealt with separately, assuming equivalent performance of NLP techniques in all language models. Although the translation approach is efficient in this study, a degree of error in terms of inaccurate translations is undeniable.

Furthermore, in automatic extraction of decision models from documents employing the extracted categories to extract the decision logic can be of research interest. While the methodology presented in chapter 6 is able to detect the decision input and decision task by enhanced feature selection and supervised machine learning, extracting decision logic is yet to be discovered. Since decision logic is embedded in a decision model in different forms of decision tables, business rules and structures, investigating actual tables and text embedded tables requires separate approaches and further discussion over the characteristics of text-based logic.

Bibliography

[1] H. P. Luhn, "A business intelligence system," *IBM Journal of Research and Development*, vol. 2, no. 4, pp. 314–319, 1958.

[2] J. R. Mashey, "Big Data and the next wave of infras-tress," in *Computer Science Division Seminar, University of California, Berkeley*, 1997.

[3] R. Baeza-Yates, B. Ribeiro-Neto, *et al.*, *Modern information retrieval.* ACM press New York, 1999.

[4] A.-H. Tan *et al.*, "Text mining: The state of the art and the challenges," in *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, vol. 8, pp. 65–70, sn, 1999.

[5] V. Gupta, G. S. Lehal, *et al.*, "A survey of text mining techniques and applications," *Journal of emerging technologies in web intelligence*, vol. 1, no. 1, pp. 60–76, 2009.

[6] J. R. Hobbs, D. E. Walker, and R. A. Amsler, "Natural language access to structured text," in *Proceedings of the 9th conference on Computational linguistics-Volume 1*, pp. 127–132, Academia Praha, 1982.

[7] A. Hotho, A. Nürnberger, and G. Paaß, "A brief survey of text mining," in *Ldv Forum*, vol. 20, pp. 19–62, Citeseer, 2005.

[8] B. S. Manjunath and W.-Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 8, pp. 837–842, 1996.

[9] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 275–281, ACM, 1998.

[10] S. Soderland, "Learning information extraction rules for semi-structured and free text," *Machine learning*, vol. 34, no. 1-3, pp. 233–272, 1999.

[11] G. G. Chowdhury, “Natural language processing,” *Annual review of information science and technology*, vol. 37, no. 1, pp. 51–89, 2003.

[12] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman, “Using latent semantic analysis to improve access to textual information,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 281–285, Acm, 1988.

[13] F. Sebastiani, “Machine learning in automated text categorization,” *ACM computing surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.

[14] A. Ritter, S. Clark, O. Etzioni, *et al.*, “Named entity recognition in tweets: an experimental study,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1524–1534, Association for Computational Linguistics, 2011.

[15] B. Liu, “Sentiment analysis and subjectivity,” *Handbook of natural language processing*, vol. 2, pp. 627–666, 2010.

[16] E. Gabrilovich and S. Markovitch, “Computing semantic relatedness using wikipedia-based explicit semantic analysis,” in *IJcAI*, vol. 7, pp. 1606–1611, 2007.

[17] K. Lang, “Newsweeder: Learning to filter netnews,” in *Proceedings of the 12th international conference on machine learning*, vol. 10, pp. 331–339, 1995.

[18] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, “Building a large annotated corpus of english: The penn treebank,” *Computational linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[19] E. Haddi, X. Liu, and Y. Shi, “The role of text pre-processing in sentiment analysis,” *Procedia Computer Science*, vol. 17, pp. 26–32, 2013.

[20] V. Srividhya and R. Anitha, “Evaluating preprocessing techniques in text categorization,” *International journal of computer science and application*, vol. 47, no. 11, pp. 49–51, 2010.

[21] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” *Machine learning: ECML-98*, pp. 137–142, 1998.

[22] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: sentiment classification using machine learning techniques,” in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79–86, Association for Computational Linguistics, 2002.

[23] N. Slonim and N. Tishby, "The power of word clusters for text classification," in *23rd European Colloquium on Information Retrieval Research*, vol. 1, p. 200, 2001.

[24] D. D. Lewis and M. Ringuette, "A comparison of two learning algorithms for text categorization," in *Third annual symposium on document analysis and information retrieval*, vol. 33, pp. 81–93, 1994.

[25] E. Wiener, J. O. Pedersen, A. S. Weigend, *et al.*, "A neural network approach to topic spotting," in *Proceedings of SDAIR-95, 4th annual symposium on document analysis and information retrieval*, vol. 317, p. 332, Las Vegas, NV, 1995.

[26] H. Schütze, D. A. Hull, and J. O. Pedersen, "A comparison of classifiers and document representations for the routing problem," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 229–237, ACM, 1995.

[27] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Icml*, vol. 97, pp. 412–420, 1997.

[28] A. Yarahmadi, M. Creemers, H. Qabbaah, and K. Vanhoof, "Unraveling bilingual multi-feature based text classification: A case study," *International Journal of Information Theories and Applications*, vol. 24, no. 4, pp. 3–18, 2017.

[29] A. McCallum, R. Rosenfeld, T. M. Mitchell, and A. Y. Ng, "Improving text classification by shrinkage in a hierarchy of classes.," in *ICML*, vol. 98, pp. 359–367, 1998.

[30] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, pp. 649–657, 2015.

[31] P. Wang and C. Domeniconi, "Building semantic kernels for text classification using wikipedia," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 713–721, ACM, 2008.

[32] P. Russom, "Bi search and text analytics," *TDWI Best Practices Report*, pp. 9–11, 2007.

[33] S. Dasgupta and V. Ng, "Topic-wise, sentiment-wise, or otherwise?: Identifying the hidden dimension for unsupervised text classification," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2*, pp. 580–589, Association for Computational Linguistics, 2009.

[34] K. Kira and L. A. Rendell, “A practical approach to feature selection,” in *Proceedings of the ninth international workshop on Machine learning*, pp. 249–256, 1992.

[35] I. S. Dhillon, S. Mallela, and R. Kumar, “A divisive information-theoretic feature clustering algorithm for text classification,” *Journal of machine learning research*, vol. 3, no. Mar, pp. 1265–1287, 2003.

[36] L. Liu, J. Kang, J. Yu, and Z. Wang, “A comparative study on unsupervised feature selection methods for text clustering,” in *Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE’05. Proceedings of 2005 IEEE International Conference on*, pp. 597–601, IEEE, 2005.

[37] N. Wiratunga, I. Koychev, and S. Massie, “Feature selection and generalisation for retrieval of textual cases,” in *European Conference on Case-Based Reasoning*, pp. 806–820, Springer, 2004.

[38] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[39] S. Robertson, “Understanding inverse document frequency: on theoretical arguments for idf,” *Journal of documentation*, vol. 60, no. 5, pp. 503–520, 2004.

[40] R. Mitkov, *The Oxford handbook of computational linguistics.* Oxford University Press, 2005.

[41] J. H. Martin and D. Jurafsky, “Speech and language processing,” *International Edition*, vol. 710, p. 25, 2000.

[42] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, “Feature-rich part-of-speech tagging with a cyclic dependency network,” in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pp. 173–180, Association for Computational Linguistics, 2003.

[43] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets.* Cambridge university press, 2014.

[44] C. J. Fox, “Lexical analysis and stoplists.,” 1992.

[45] J. Xu, A. Fraser, and R. Weischedel, “Empirical studies in strategies for arabic retrieval,” in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 269–274, ACM, 2002.

[46] M. F. Porter, “An algorithm for suffix stripping,” *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[47] C. J. Van Rijsbergen, S. E. Robertson, and M. F. Porter, *New models in probabilistic information retrieval.* British Library Research and Development Department London, 1980.

[48] W. B. Cavnar, J. M. Trenkle, *et al.*, "N-gram-based text categorization," *Ann arbor mi*, vol. 48113, no. 2, pp. 161–175, 1994.

[49] P. Mcnamee and J. Mayfield, "Character n-gram tokenization for european language text retrieval," *Information retrieval*, vol. 7, no. 1-2, pp. 73–97, 2004.

[50] E. Millar, D. Shen, J. Liu, and C. Nicholas, "Performance and scalability of a large-scale n-gram based information retrieval system," *Journal of digital information*, vol. 1, no. 5, 2000.

[51] X. Wang, A. McCallum, and X. Wei, "Topical n-grams: Phrase and topic discovery, with an application to information retrieval," in *icdm*, pp. 697–702, IEEE, 2007.

[52] C. C. Aggarwal and C. Zhai, *Mining text data.* Springer Science & Business Media, 2012.

[53] L. AlSumait, D. Barbará, and C. Domeniconi, "On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pp. 3–12, IEEE, 2008.

[54] C. Goddard, *Semantic analysis: A practical introduction.* Oxford University Press, 2011.

[55] N. Jindal and B. Liu, "Review spam detection," in *Proceedings of the 16th international conference on World Wide Web*, pp. 1189–1190, ACM, 2007.

[56] A. McCallum, K. Nigam, J. Rennie, and K. Seymore, "A machine learning approach to building domain-specific search engines," in *IJCAI*, vol. 99, pp. 662–667, 1999.

[57] D. Fensel, Y. Ding, B. Omelayenko, E. Schulten, G. Botquin, M. Brown, and A. Flett, "Product data integration in b2b e-commerce," *IEEE Intelligent Systems*, vol. 16, no. 4, pp. 54–59, 2001.

[58] C. Lee and G. G. Lee, "Information gain and divergence-based feature selection for machine learning-based text categorization," *Information processing & management*, vol. 42, no. 1, pp. 155–165, 2006.

[59] J. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 335–336, ACM, 1998.

[60] D. Guru and M. Suhil, "A novel term_class relevance measure for text categorization," *Procedia Computer Science*, vol. 45, pp. 13–22, 2015.

[61] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[62] Z. Zheng, X. Wu, and R. Srihari, "Feature selection for text categorization on imbalanced data," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 80–89, 2004.

[63] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[64] C. Breitinger, B. Gipp, and S. Langer, "Research-paper recommender systems: a literature survey," *International Journal on Digital Libraries*, vol. 17, no. 4, pp. 305–338, 2015.

[65] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[66] G. A. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[67] T. Pedersen, S. Patwardhan, and J. Michelizzi, "WordNet: Similarity: measuring the relatedness of concepts," in *Demonstration papers at HLT-NAACL 2004*, pp. 38–41, Association for Computational Linguistics, 2004.

[68] S. Shankar and I. Lin, "Applying machine learning to product categorization," *Department Of Computer Science, Stanford University*, 2011.

[69] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3–24, 2007.

[70] N. Ur-Rahman and J. A. Harding, "Textual data mining for industrial knowledge management and text classification: A business oriented approach," *Expert Systems with Applications*, vol. 39, no. 5, pp. 4729–4739, 2012.

[71] R. Ghani, K. Probst, Y. Liu, M. Krema, and A. Fano, "Text mining for product attribute extraction," *ACM SIGKDD Explorations Newsletter*, vol. 8, no. 1, pp. 41–48, 2006.

[72] A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Natural language processing and text mining*, pp. 9–28, Springer, 2007.

[73] D. Beneventano, F. Guerra, S. Magnani, and M. Vincini, "A web service based framework for the semantic mapping between product classification schemas," *Journal of Electronic Commerce Research*, vol. 5, pp. 114–127, 2004.

[74] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval.* Cambridge University Press, 2008.

[75] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242, pp. 133–142, 2003.

[76] S. Vijayarani, M. J. Ilamathi, and M. Nithya, "Preprocessing techniques for text mining-an overview," *International Journal of Computer Science & Communication Networks*, vol. 5, no. 1, pp. 7–16, 2015.

[77] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," in *Mining text data*, pp. 163–222, Springer, 2012.

[78] J. Rennie, L. Shih, J. Teevan, and D. Karger, "Tackling the poor assumptions of naive bayes classifiers (pdf)," ICML, 2003.

[79] T. Zhang and F. J. Oles, "Text categorization based on regularized linear classification methods," *Information retrieval*, vol. 4, no. 1, pp. 5–31, 2001.

[80] F. Lu and Q. Bai, "A refined weighted k-nearest neighbors algorithm for text categorization," in *Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference on*, pp. 326–330, IEEE, 2010.

[81] B. Hofner, A. Mayr, N. Robinzonov, and M. Schmid, "Model-based boosting in r: a hands-on tutorial using the r package mboost," *Computational Statistics*, vol. 29, no. 1-2, pp. 3–35, 2014.

[82] J. Friedman, T. Hastie, and R. Tibshirani, "GLMNET: Lasso and elastic-net regularized generalized linear models," *R package version*, vol. 1, no. 4, 2009.

[83] J. Elith, S. J. Phillips, T. Hastie, M. Dudík, Y. E. Chee, and C. J. Yates, "A statistical explanation of maxent for ecologists," *Diversity and distributions*, vol. 17, no. 1, pp. 43–57, 2011.

[84] J. D. Mcauliffe and D. M. Blei, "Supervised topic models," in *Advances in neural information processing systems*, pp. 121–128, 2008.

[85] T. Kudo and Y. Matsumoto, "A boosting algorithm for classification of semi-structured text," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.

[86] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine learning*, vol. 39, no. 2-3, pp. 135–168, 2000.

[87] K. Nigam, J. Lafferty, and A. McCallum, "Using maximum entropy for text classification," in *IJCAI-99 workshop on machine learning for information filtering*, vol. 1, pp. 61–67, 1999.

[88] S. Della Pietra, V. Della Pietra, and J. Lafferty, "Inducing features of random fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 4, pp. 380–393, 1997.

[89] H. L. Chieu and H. T. Ng, "Named entity recognition: a maximum entropy approach using global information," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pp. 1–7, Association for Computational Linguistics, 2002.

[90] N. Mehra, S. Khandelwal, and P. Patel, "Sentiment identification using maximum entropy analysis of movie reviews," *St anford Univer sity, USA in*, 2002.

[91] A. Ratnaparkhi, "A maximum entropy model for part-of-speech tagging," in *Conference on Empirical Methods in Natural Language Processing*, 1996.

[92] T. P. Jurka, L. Collingwood, A. E. Boydstun, E. Grossman, and W. van Atteveldt, "RTextTools: Automatic text classification via supervised learning," *R package version*, vol. 1, no. 9, 2012.

[93] A. E. Boydstun, *Making the news: Politics, the media, and agenda setting.* University of Chicago Press, 2013.

[94] S. Arlot, A. Celisse, *et al.*, "A survey of cross-validation procedures for model selection," *Statistics surveys*, vol. 4, pp. 40–79, 2010.

[95] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[96] A. Huang, "Similarity measures for text document clustering," in *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, pp. 49–56, 2008.

[97] V. Vapnik, *Statistical learning theory. 1998.* Wiley, New York, 1998.

[98] A. Kaur and D. Chopra, "Comparison of text mining tools," in *Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), 2016 5th International Conference on*, pp. 186–192, IEEE, 2016.

[99] S. Lesser, "Coding of regulatory texts for the creation of decision models," *university of st. andrews, school of computer science*, 2016.

[100] D. D. Lewis, “Text representation for intelligent text retrieval: A classification-oriented view,” *Text-based intelligent systems: current research and practice in information extraction and retrieval*, pp. 179–197, 1992.

[101] T. Gonçalves and P. Quaresma, “Is linguistic information relevant for the classification of legal texts?,” in *Proceedings of the 10th international conference on Artificial intelligence and law*, pp. 168–176, ACM, 2005.

[102] C.-W. Hsu, C.-C. Chang, C.-J. Lin, *et al.*, “A practical guide to support vector classification,” 2003. Technical Report, Department of Computer Science and Information Engineering, University of National Taiwan, Taipei, pp. 1-12.

[103] A. Ratner, “Leveraging document structure for better classification of complex legal documents.” http://cs229.stanford.edu/proj2014/Alex%20Ratner,%20Classifying%20Complex%20Legal%20Documents.pdf/, 2014.

[104] A. Wyner, R. Mochales-Palau, M.-F. Moens, and D. Milward, “Approaches to text mining arguments from legal cases,” in *Semantic processing of legal texts*, pp. 60–79, Springer, 2010.

[105] B. Hachey and C. Grover, “Sentence classification experiments for legal text summarisation,” in *Proc. 17th Annual Conference on Legal Knowledge and Information Systems (Jurix-2004)*, pp. 29–38, 2004.

[106] H. P. Luhn, “A statistical approach to mechanized encoding and searching of literary information,” *IBM Journal of research and development*, vol. 1, no. 4, pp. 309–317, 1957.

[107] H. Cunningham, “Gate, a general architecture for text engineering,” *Computers and the Humanities*, vol. 36, no. 2, pp. 223–254, 2002.

[108] T. Muhr and S. Friese, “User’s manual for atlas. ti 5.0,” *Berlin: ATLAS. ti Scientific Software Development GmbH*, 2004.

[109] S. Bird, “Nltk: the natural language toolkit,” in *Proceedings of the COLING/ACL on Interactive presentation sessions*, pp. 69–72, Association for Computational Linguistics, 2006.

[110] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, “Scikit-learn: Machine learning in python,” *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[111] A. McCallum, K. Nigam, *et al.*, “A comparison of event models for naive bayes text classification,” in *AAAI-98 workshop on learning for text categorization*, vol. 752, pp. 41–48, Madison, WI, 1998.

[112] A. Jalilvand and N. Salim, "Feature unionization: A novel approach for dimension reduction," *Applied Soft Computing*, vol. 52, pp. 1253–1261, 2017.

[113] D. Chen and C. Manning, "A fast and accurate dependency parser using neural networks," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 740–750, 2014.

[114] M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger, "The penn treebank: annotating predicate argument structure," in *Proceedings of the workshop on Human Language Technology*, pp. 114–119, Association for Computational Linguistics, 1994.

[115] X. Tang and J. Cao, "Automatic genre classification via n-grams of part-of-speech tags," *Procedia-Social and Behavioral Sciences*, vol. 198, pp. 474–478, 2015.

[116] S. Bloehdorn and A. Hotho, "Text classification by boosting weak learners based on terms and concepts," in *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pp. 331–334, IEEE, 2004.

[117] B. Xu, X. Guo, Y. Ye, and J. Cheng, "An improved random forest classifier for text categorization.," *JCP*, vol. 7, no. 12, pp. 2913–2920, 2012.

[118] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Appendices

Appendix A: Penn Part-of-Speech Tags Manual

Penn part-of-speech Treebank		
Number	Tag	Description
1	CC	Coordinating conjunction
2	CD	Cardinal number
3	DT	Determiner
4	EX	Existential *there*
5	FW	Foreign word
6	IN	Preposition or subordinating conjunction
7	JJ	Adjective
8	JJR	Adjective, comparative
9	JJS	Adjective, superlative
10	LS	List item marker
11	MD	Modal
12	NN	Noun, singular or mass
13	NNS	Noun, plural
14	NNP	Proper noun, singular
15	NNPS	Proper noun, plural
16	PDT	Predeterminer
17	POS	Possessive ending
18	PRP	Personal pronoun
19	PRP$	Possessive pronoun
20	RB	Adverb
21	RBR	Adverb, comparative
22	RBS	Adverb, superlative
23	RP	Particle
24	SYM	Symbol
25	TO	*to*
26	UH	Interjection
27	VB	Verb, base form
28	VBD	Verb, past tense
29	VBG	Verb, gerund or present participle
30	VBN	Verb, past participle
31	VBP	Verb, non-3rd person singular present
32	VBZ	Verb, 3rd person singular present
33	WDT	Wh-determiner
34	WP	Wh-pronoun
35	WP$	Possessive wh-pronoun
36	WRP	Wh-adverb

Appendix B: NLTK English Stopwords

['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]

Appendix C: Code of Conduct Samples

Apple Supplier Code of Conduct

Apple is committed to the highest standards of social and environmental responsibility and ethical conduct. Apple's suppliers are required to provide safe working conditions, treat workers with dignity and respect, act fairly and ethically, and use environmentally responsible practices wherever they make products or perform services for Apple. Apple requires its suppliers to operate in accordance with the principles in this Apple Supplier Code of Conduct (”Code”) and in full compliance with all applicable laws and regulations. This Code goes beyond mere compliance with the law by drawing upon internationally recognized standards to advance social and environmental responsibility. When differences arise between standards and legal requirements, the stricter standard shall apply, in compliance with applicable law. This Code outlines Apple's expectations for Supplier conduct regarding labor and human rights, health and safety, environmental protection, ethics, and management practices.

Apple will assess its suppliers' compliance with this Code, and any violations of this Code may jeopardize the supplier's business relationship with Apple, up to and including termination. This Code applies to Apple suppliers and their subsidiaries, affiliates, and subcontractors (each a ”Supplier”) providing goods or services to Apple, or for use in or with Apple products.

Additionally, Apple maintains detailed standards that explicitly define our expectations for compliance with this Code.

Labor and Human Rights

Apple believes all workers in our supply chain deserve a fair and ethical workplace. Workers must be treated with the utmost dignity and respect, and suppliers shall uphold the highest standards of human rights.

Anti-discrimination

Supplier shall not discriminate against any worker based on age, disability, ethnicity, gender, marital status, national origin, political affiliation, race, religion, sexual orientation, gender identity, union membership, or any other status protected by country law, in hiring and other employment practices. Supplier shall not require pregnancy or medical tests, except where required by applicable laws or regulations or prudent for workplace safety, and shall not improperly discriminate based on test results.

Anti-Harassment and Abuse

Supplier shall commit to a workplace free of harassment and abuse. Supplier shall not threaten workers with, or subject them to, harsh or inhumane treatment, including but not limited to verbal abuse and harassment, psychological harassment, mental and physical coercion, and sexual harassment.

Prevention of Involuntary Labor and Human Trafficking

Supplier shall ensure that all work is voluntary. Supplier shall not traffic persons or use any form of slave, forced, bonded, indentured, or prison labor. Involuntary labor includes the transportation, harboring, recruitment, transfer, receipt, or employment of persons by means of threat, force, coercion, abduction, fraud, or payments to any person having control over another person for the purpose of exploitation.
Supplier shall not withhold workers' original government-issued identification and travel documents. Supplier shall ensure that workers' contracts clearly convey the conditions of employment in a language understood by the workers. Supplier shall not impose unreasonable restrictions on movement within the workplace or upon entering or exiting company-provided facilities.

Supplier shall ensure that the third-party recruitment agencies it uses are compliant with the provisions of this Code and the law. Suppliers recruiting foreign contract workers either directly or through third party agencies shall be responsible for payment of all recruitment-related fees and expenses.

Prevention of Underage Labor

Supplier shall employ only workers who are at least 15 years of age or the applicable minimum legal age, whichever is higher. Supplier may provide legitimate workplace apprenticeship programs for educational benefit that are consistent with Article 6 of ILO Minimum Age Convention No. 138 or light work consistent with Article 7 of ILO Minimum Age Convention No. 138.

Juvenile Worker Protections

Supplier may employ juveniles who are older than the applicable legal minimum age but are younger than 18 years of age, provided they do not perform work that might jeopardize their heath, safety, or morals, consistent with ILO Minimum Age Convention No. 138. Supplier shall not require juvenile workers to work overtime or perform night work.

Student Worker Protections

Supplier shall ensure proper management of student workers through proper maintenance of student records, rigorous due diligence of educational partners, and protection of students' rights in accordance with applicable law and regulations. Supplier shall provide appropriate support and training to all student workers.

Working Hours

A workweek shall be restricted to 60 hours, including overtime, and workers shall take at least one day off every seven days except in emergencies or unusual situations. Regular work week shall not exceed 48 hours. Supplier shall follow all applicable laws and regulations with respect to working hours and days of rest, and all overtime must be voluntary.

Wages and Benefits

Supplier shall ensure that all workers receive at least the legally mandated minimum wages and benefits. Supplier shall offer vacation time, leave periods, and time off for legally recognized holidays.

Supplier shall compensate workers for overtime hours at the legal premium rate. Supplier shall communicate pay structure and pay periods to all workers. Supplier shall pay accurate wages in a timely manner, and wage deductions shall not be used as a disciplinary measure. All use of temporary and outsourced labor will be within the limits of the local law.

Freedom of Association and Collective Bargaining

Supplier shall freely allow workers to associate with others, form, and join (or refrain from joining) organizations of their choice, and bargain collectively, without interference, discrimination, retaliation, or harassment. In the absence of formal representation, Supplier shall ensure that workers have a mechanism to report grievances and that facilitates open communication between management and workers.

Health and Safety

Worker health, safety, and well-being is important to Apple. Supplier shall provide and maintain a safe work environment and integrate sound health and safety management practices into its business. Workers shall have the right to refuse unsafe work and to report unhealthy working conditions.

Occupational Health, Safety, and Hazard Prevention

Supplier shall identify, evaluate, and manage occupational health and safety hazards through a prioritized process of hazard elimination, engineering controls, and/or administrative controls. Supplier shall provide workers with job-related, appropriately maintained personal protective equipment and instruction on its proper use.

Emergency Prevention, Preparedness, and Response

Supplier shall identify and assess potential emergency situations. For each situation, Supplier shall develop and implement emergency plans and response procedures that minimize harm to life, environment, and property.To the extent that Supplier transports goods for Apple into the United States, Supplier shall comply with the C-TPAT (Customs-Trade Partnership Against Terrorism) security procedures on the U.S. Customs website at www.cbp.gov (or other website established for such purpose by the U.S. government).

Incident Management

Supplier shall have a system for workers to report health and safety incidents and near-misses, as well as a system to investigate, track, and manage such reports. Supplier shall implement corrective action plans to mitigate risks, provide necessary medical treatment, and facilitate workers' return to work.

Ergonomics

Supplier shall identify, evaluate, and control worker exposure to tasks that pose ergonomic risk such as excessive force, improper lifting positions, or repetitiveness. Supplier shall integrate this process into the qualification of all new or modified production lines, equipment, tools, and workstations.

Working and Living Conditions

Supplier shall provide workers with reasonably accessible and clean toilet facilities and potable water. Supplier-provided dining, food preparation, and storage facilities shall be sanitary. Worker dormitories provided by Supplier or a third-party shall be clean and safe and provide reasonable living space.

Health and Safety Communication

Supplier shall provide workers with appropriate workplace health and safety training in their primary language. Health and safety related information shall be clearly posted in the facility.

Worker Health and Safety Committees

Supplier is encouraged to initiate and support worker health and safety committees to enhance ongoing health and safety education and to encourage worker input on, and participation in, health and safety issues in the workplace.

Environment

Apple is committed to protecting the environment, and environmental responsibility is at the core of how we operate. Supplier shall develop, implement, and maintain environmentally responsible business practices.

Hazardous Substance Management and Restriction

Supplier shall implement a systematic approach to identify, manage, reduce, and responsibly dispose of or recycle hazardous substances. Supplier shall comply with Apple's Regulated Substances Specification for all goods it manufactures for and provides to Apple.

Non-Hazardous Waste Management

Supplier shall implement a systematic approach to identify, manage, reduce, and responsibly dispose of or recycle non-hazardous waste.

Waste-water Management

Supplier shall implement a systematic approach to identify, control, and reduce wastewater produced by its operations. Supplier shall conduct routine monitoring of the performance of its wastewater treatment systems.

Storm-water Management

Supplier shall implement a systematic approach to prevent contamination of stormwater runoff. Supplier shall prevent illegal discharges and spills from entering storm drains.

Air Emissions Management

Supplier shall identify, manage, reduce, and responsibly control air emissions emanating from its operations that pose a hazard to the environment. Supplier shall conduct routine monitoring of the performance of its air emission control systems.

Boundary Noise

Supplier shall identify, control, monitor, and reduce noise generated by the facility that affects boundary noise levels.

Environmental Permits and Reporting

Supplier shall obtain, keep current, and comply with all required environmental permits. Supplier shall comply with the reporting requirements of applicable permits and regulations.

Pollution Prevention and Resource Reduction

Supplier shall reduce energy, water, and natural resource consumption by implementing conservation and substitution measures. Supplier shall minimize hazardous substances consumption by implementing reduction and substitution measures.

Ethics

Apple expects the highest standards of ethical conduct in all of our endeavors. Supplier shall always be ethical in every aspect of its business, including relationships, practices, sourcing, and operations.

Business Integrity

Supplier shall not engage in corruption, extortion, embezzlement, or bribery to obtain an unfair or improper advantage. Supplier shall abide by all applicable anti-corruption laws and regulations of the countries in which it operates, including the Foreign Corrupt Practices Act (FCPA) and applicable international anti-corruption conventions.

Disclosure of Information

Supplier shall accurately record information regarding its business activities, labor, health and safety, and environmental practices and shall disclose such information, without falsification or misrepresentation, to all appropriate parties.

Protection of Intellectual Property

Supplier shall respect intellectual property rights and safeguard customer information. Supplier shall manage technology and know-how in a manner that protects intellectual property rights.

Whistle-blower Protection and Anonymous Complaints

Supplier shall provide an anonymous complaint mechanism for managers and workers to report workplace grievances. Supplier shall protect whistle-blower confidentiality and prohibit retaliation.

Community Engagement

Supplier is encouraged to help foster social and economic development and contribute to the sustainability of the communities in which it operates.

Responsible Sourcing of Minerals

Supplier shall exercise due diligence, in accordance with the OECD Due Diligence Guidance for Responsible Supply Chains of Minerals from Conflict-Affected and High-Risk Areas, on its entire supply chain with respect to the sourcing of all tin, tantalum, tungsten, and gold contained in its products, to determine whether those metals are from the Democratic Republic of the Congo (”DRC”) or any adjoining country and, if so, to determine whether those metals directly or indirectly financed or benefited armed groups that are perpetrators of serious human rights abuses in the DRC or an adjoining country. Countries that adjoin the DRC are Angola, Burundi, Central African Republic, the Republic of the Congo, Rwanda, South Sudan, Tanzania, Uganda, and Zambia.

Management Commitment

Apple believes that sound management systems and commitment are key to enriching the social and environmental well-being of our supply chain. Apple holds its suppliers accountable to this Code and all of its standards. Supplier shall implement or maintain, as applicable, a management system that facilitates compliance with this Code and the law, identifies and mitigates related operational risks, and facilitates continuous improvement.

Company Statement

Supplier shall develop a company statement affirming its commitment to high standards of social and environmental responsibility, ethical conduct, and continuous improvement. Supplier shall post this statement in the primary local language at all of its facilities.

Management Accountability and Responsibility

Supplier shall identify company representatives responsible for ensuring implementation and periodic review of its management systems. Supplier shall have a Corporate Social Responsibility (CSR) or Sustainability representative that reports directly to executive management and has the responsibility and authority to manage social and environmental compliance requirements for the business.

Risk Assessment and Management

Supplier shall develop and maintain a process to identify labor and human rights, health and safety, environmental, business ethics, and legal compliance risks associated with its operations; determine the relative significance of each risk; and implement appropriate procedures and controls to control the identified risks.

Performance Objectives with Implementation Plans and Measures

Supplier shall have written standards, performance objectives, targets, and implementation plans, including periodic assessments of the performance against those objectives.

Audits and Assessments

Supplier shall perform periodic evaluations of its facilities and operations, and the facilities and operations of its subcontractors and next-tier suppliers to ensure compliance with this Code and the law.
Supplier will permit Apple and/or a third party designated by Apple to periodically evaluate Supplier's facilities and operations, and those of its subcontractors and next-tier suppliers, to the extent they are providing goods or services to Apple, for Apple's benefit, or for use in Apple products.

Documentation and Records

Supplier shall have processes to identify, understand, and implement applicable laws and regulations and requirements of this Code. Suppliers shall maintain documents and records to ensure regulatory compliance.

Training and Communication

Supplier shall develop and maintain management and worker training programs to facilitate proper implementation of its policies and procedures and to fulfill Supplier's continuous improvement objectives. Supplier shall have a process for communicating clear and accurate information about its performance, practices, policies, and expectations to its workers, next-tier supplier(s), and customers. Supplier shall have an ongoing process to obtain feedback on its practices related to this Code and to foster continuous improvement.

Corrective Action Process

Supplier shall have a process for timely correction of any deficiencies or violations identified by an internal or external audit, assessment, inspection, investigation, or review.

Lidl Code of Conduct

Within its own company and in business relations with its business partners, Lidl pays heed to the implementation of minimum social standards. Lidl is member of the “Business Social Compliance Initiative” (BSCI) with the European Foreign Trade Association. On this basis, Lidl has developed its own Code of Conduct by which Lidl pursues the aim of improving minimum social standards with its business partners in the various countries. These minimum standards are the major foundation of business relations of Lidl with its contracting partners.

Human Dignity

Human dignity is to be respected as elementary requirement for living together.

Compliance with legal regulations

Compliance is to be assured with the valid national and other relevant laws and regulations as well as with the Conventions of the ILO and the UN. Out of all the applicable regulations, those that are best suited to provide the protection aimed at shall always be definitive. Bribery, corruptibility and other corruption are prohibited.

Ban of Child Labor

For the production of goods or the provision of services for Lidl, child labor such as defined by the ILO and UN Conventions, the international standard SA8000 or by national regulations is prohibited. Breaches of this ban shall be rectified by documented strategies and procedures. The schooling of children is to be given the appropriate support. Young adults (adolescents) who are at least 15 and not yet 18 years of age as defined in the international standard SA8000 may not be employed outside of school hours. Daily working hours shall under no circumstances exceed eight hours and the total period of time spent daily at school, at work and during transport shall not exceed 10 hours. Young adults (adolescents) shall not be permitted to work at night.

Ban of Forced Labour and Disciplinary Measures

All and any form of forced labour is prohibited. The application of physical punishment, mental or physical coercion and verbal insulting abuse is prohibited.

Working Conditions and Payment

Compliance is to be upheld with nationally valid regulations governing labor law. The wages and other allowances must at least comply with the statutory regulations and/or the standards of the local production industry. The wages and other benefits are to be clearly defined and paid out and/or made regularly. The aim is to pay wages and other allowances that cover the cost of living, insofar as the statutory minimum wages are inadequate in this respect. Deductions for non-cash benefits are only allowed to a minimal extent and only in relation to the value of the non-cash benefit. Regular maximum working hours are subject to statutory regulations. They amount to no more than 48 hours per week. The number of hours' overtime is no more than 12 hours per week; any extra hours of overtime are only permissible if necessitated at short notice for operational reasons and if permitted by a collective regulation. Hours of overtime are paid for separately or compensated by leisure hours. After 6 consecutive workdays, an employee shall be entitled to one work-free day. More consecutive workdays are only allowed if this is permitted by national law and a collective ruling.

Ban of Discrimination

Discrimination on the grounds of gender or sexual identity, age, religion or philosophy, race, ethnic origin, national or social origin or an employee's disablement is prohibited.

Freedom to organize and hold meetings

The rights of the employees to set up labour organizations and to become members of them as well as the right to take joint action under the respective national laws and regulations as well as the ILO Conventions are not allowed to be restricted. Employees must not be discriminated against if they exercise these rights.

Occupational Health and Safety

Safe conditions at the workplace that are compatible with health are to be guaranteed. Conditions at the workplace and in company facilities and working conditions that violate fundamental human rights are prohibited. In particular, young adults (adolescents) should not be exposed to hazardous, unsafe or unhealthy situations that endanger their health and development. The personnel are to be regularly briefed on health and safety at the workplace.
An officer for occupational health and safety is to be appointed from the area of Management. He is responsible for the introduction of and compliance with health and safety standards at the workplace.

Environmental Protection

Compliance with environmental and safety regulations governing the treatment of waste, the handling of chemicals or other dangerous materials or substances is to be assured. The employees are to be briefed on handling dangerous materials and substances.

In-company Implementation

An in-company strategy of social responsibility and appropriate in-company procedures are to implement and support the aforementioned social standards. A company monitoring system for breaches of these social standards shall be set up; employees who give notification of such breaches through the system must not be penalized or placed at a disadvantage as a result. The business partners agree that the implementation of the social standards may be controlled at any time, either by Lidl itself or by an impartial controller appointed by Lidl.

Each of Lidl's contracting partners declares its readiness to implement these social standards in its respective company and also to impose them on its own business partners and ensure they are implemented.

Appendix D: Initial annotated training set

Document		
Document number	Content	Proposed class
1	If you believe that your own or another employee's behavior contravenes the values and principles of conduct outlined in this Code	condition
2	if you have a genuine concern that something is not right	condition
3	If you have questions or concerns about how this Code or Citi's policies apply to you or others	condition
4	If there appears to be a conflict between this Code and local laws, or if you have questions regarding the interpretation of applicable laws	condition
5	If such policies differ from Citi policies governing the same topic	condition
6	If you are unclear as to your responsibilities	condition
7	If you are in doubt as to your responsibilities under the Fraud Management Policy	condition
8	If you have reason to believe that any Citi employee, or anyone working on our Company's behalf, may have engaged in misconduct	condition
9	If you encounter a situation that does not feel quite right	condition
10	When faced with questions beyond those addressed in the Code	condition

Table 1: Annotated document sample of training dataset part 1

Document		
Document number	Content	Proposed class
11	If you have any questions regarding the best course of action in a particular situation	condition
12	if you reasonably suspect or become aware of a possible violation of a law, regulation, Citi policy or ethical standard	condition
13	If you are uncomfortable about raising your concerns with the contacts listed	condition
14	if you do choose to remain anonymous	condition
15	If you raise an ethical issue and you do not believe the issue has been addressed	condition
16	If something feels unethical or improper to you	condition
17	If you have any questions about the best course of action in a particular situation	condition
18	if you suspect or become aware of a possible violation of law, regulation, Citi policy or ethical standard	condition
19	If you believe that you are being subjected to discrimination or harassment	condition
20	or if you observe or receive a complaint regarding such behavior	condition

Table 2: Annotated document sample of training dataset part 2

Document		
Document number	Content	Proposed class
21	If you receive an inappropriate e-mail from another employee	condition
22	If you expect to pay legal expenses to defend yourself in a civil or criminal action, suit or proceeding arising from your service as an officer, director or employee of Citi	condition
23	If management determines, based on governing documents and applicable law, that you are entitled to representation, and for any reason a Citi-designated attorney cannot represent you	condition
24	If such an error occur	condition
25	In addition, once your employment or association with Citi ends	condition
26	If a representative from the media contacts you seeking a statement on behalf of Citi”	condition
27	when permitted under applicable law if they are (1) nominal in value (i.e., less than or equal to USD 100 per provider per calendar year) and (2) appropriate, customary and reasonable gifts based on family or personal relationships, and clearly not meant to influence Citi business”	condition
28	when permitted under applicable law and in accordance with Citi policies	condition
29	If you receive a gift without an opportunity to refuse it	condition
30	If you have any questions about the appropriateness of accepting a gift, invitation, raffle or other prize	condition

Table 3: Annotated document sample of training dataset part 3

Document		
Document number	Content	Proposed class
31	In accepting a position or assignment with Citi, each of us is accountable for our own behavior, including compliance with the law, this Code of Conduct, Citi's policies and the policies and procedures of our respective businesses and legal entities	obligation
32	we must never do anything to put that reputation at risk	obligation
33	You must be sensitive to any activities, interests or relationships that might interfere with, or even appear to interfere with, your ability to act in the best interests of Citi and our clients	obligation
34	Check the requirements of your specific business, legal entity and region with regard to these events and comply with any applicable restrictions	obligation
35	Such gifts must be disclosed and reported	obligation
36	Consult with your Compliance Officer and the requirements of your specific business and legal entity for further guidance	obligation
37	Politely refuse it explain that Citi policy prohibits you from accepting it	obligation
38	explain that Citi policy prohibits you from accepting it	obligation
39	consult your manager or Compliance Officer for guidance	obligation
40	You are responsible for complying with the procedures that are applicable to you	obligation

Table 4: Annotated document sample of training dataset part 4

Document		
Document number	Content	Proposed class
41	You must report gifts in accordance with any procedures your business, legal entity and region have regarding gift reporting	obligation
42	you should discuss the matter with your manager and your Compliance Officer prior to participation or acceptance	obligation
43	you must not give the gift or provide the entertainment	obligation
44	It is your responsibility to become familiar with the gift and entertainment restrictions applicable to you and to comply with all pre-approval and reporting requirements	obligation
45	Charitable contributions and charitable events funded by Citi should support Citi's philanthropic objectives and should be allocated across a variety of charitable institutions	obligation
46	Citi's workplace should be free from outside influences	obligation
47	Individual employee giving to charitable organizations should be confidential, purely voluntary, have no impact on employment or compensation decisions and be in compliance with all non-solicitation policies	obligation

Table 5: Annotated document sample of training dataset part 5

Document		
Document number	Content	Proposed class
48	You must be aware of whether your actions on behalf of Citi would create a potential conflict of interest with a client, customer or counter-party	obligation
49	In certain instances, it may be unlawful for you to engage in any transaction, class of transactions or activity that would involve or result in Citi's interests being materially adverse to the other party unless appropriate measures are taken, including the use of disclosures or information barriers	obligation
50	Please refer to the conflict of interest policies that apply to your business or region	obligation
51	All Citi employees must disclose, and receive the necessary approvals prior to participating in the following activities	obligation
52	You are also required to comply with any applicable laws, regulations and business and legal entity policies	obligation

Table 6: Annotated document sample of training dataset part 6

Document		
Document number	Content	Proposed class
53	You are responsible for identifying and raising any such activity or relationship that may pose an apparent or potential conflict of interest and to evaluate with your manager and your Compliance Officer the possible conflicts that could result	obligation
54	You should also promptly report the matter to your internal legal counsel or to the Corporate Law Department	obligation
55	you should consult your internal legal counsel, bank regulatory legal counsel or Compliance Officer for advice	obligation
56	Make it clear that you object to such a discussion	obligation
57	You should avoid all discussions that relate to pricing or price-related issues, including discounts, with any competitor	obligation
58	You are responsible for understanding and abiding by Citi policy in the countries in which you are located, as well as U.S. law	obligation
59	you should alert both your internal legal counsel and your Compliance Officer	obligation
60	you are responsible for knowing and abiding by the terms of the co-investment plan	obligation

Table 7: Annotated document sample of training dataset part 7

Document		
Document number	Content	Proposed class
61	Therefore, violations of our Code and/or law, regulation, Citi policy or procedure may result in disciplinary action up to and including termination of employment	prohibition
62	Citi prohibits any form of retaliatory action against anyone for raising concerns or questions in good faith regarding ethics, discrimination or harassment matters	prohibition
63	Citi prohibits retaliatory actions against anyone for raising concerns in good faith	prohibition
64	You should never withhold, tamper with or fail to communicate relevant information in connection with an appropriately authorized investigation	prohibition
65	We prohibit all forms of discrimination, harassment or intimidation that are unlawful or otherwise violate our policies	prohibition
66	Retaliation against individuals for raising claims of discrimination or harassment, or participating in the investigation of a claim, is also prohibited	obligation
67	In addition, do not forward any inappropriate e-mail to any external address, including to your home computer	prohibition
68	Allowing another colleague or representative to take an expected or required training for you is prohibited and may result in disciplinary action up to and including termination of employment	prohibition

Table 8: Annotated document sample of training dataset part 8

Document		
Document number	Content	Proposed class
69	Therefore, you should not have any expectation of personal privacy when you use Citi's equipment, systems and services	prohibition
70	Citi prohibits any form of retaliatory action against anyone for raising concerns or questions in good faith regarding ethics, discrimination or harassment matters	prohibition
71	You may not use Citi's equipment, systems and services for any inappropriate or unauthorized purpose or in a manner that would violate applicable law, regulation or Citi's policies or procedures	prohibition
72	Citi's intranet/Internet servers may not be used for the unauthorized downloading or use of any copyrighted or unlicensed material	prohibition
73	The Internet may not be accessed from a Citi device to view	prohibition
74	Copying, selling, using, or distributing information, software and other forms of intellectual property in violation of intellectual property laws or license agreements is prohibited	prohibition
75	You may not bring to Citi proprietary or confidential information of any former employer, or use such information to aid the business of Citi, without the prior consent of your former employer	prohibition
76	You must not disclose personal, proprietary or confidential information about any client, supplier, distributor, Citi's workforce or Citi to any unauthorized person	prohibition

Table 9: Annotated document sample of training dataset part 9

Document		
Document number	Content	Proposed class
77	Further, you may not print, download or forward such information as listed above to your home computer	prohibition
78	You are prohibited from destroying or altering any records that are potentially relevant to a violation of law	prohibition
79	Individuals cannot approve their own expenses	prohibition
80	Any false or fraudulent submission of expenses is grounds for disciplinary action up to and including termination of employment	prohibition
81	Employees may not consent to or engage in any public relations activity on behalf of Citi with clients, suppliers, distributors or others without prior approval from Global Public Affairs	prohibition
82	you may not publish, post or link to any material in written or electronic format (including books, articles, podcasts, webcasts, blogs, website postings, photos, videos or other media), make speeches, give interviews or make public appearances on behalf of or as a representative of Citi that mention Citi's operations, clients, products or services, without prior approval from your manager and the local Public Affairs Officer for your business or region	obligation
83	You must not use Citi's name, logo, trademarks or facilities for commercial purposes unrelated to your job, including outside work	prohibition
84	You may not use Citi communications, equipment, systems and services for personal use of external social media sites	prohibition

Table 10: Annotated document sample of training dataset part 10

Document		
Document number	Content	Proposed class
85	Do not infringe on Citi logos, brand names, taglines, slogans or other trademarks	prohibition
86	We must never compromise that integrity, either for personal benefit or for Citi's purported benefit	prohibition
87	Such information must not be shared or discussed outside Citi	prohibition
88	You should not discuss sensitive matters or proprietary or confidential information in public places such as elevators, hallways, restaurants, restrooms and public transportation, or on the Internet or any other electronic media	prohibition
89	Failure to immediately report the above is a serious offense and may result in disciplinary action up to and including termination of employment	prohibition
90	Misusing controlled substances or selling, manufacturing, distributing, possessing, using or being under the influence of illegal drugs or alcohol, is prohibited in the workplace	prohibition

Table 11: Annotated document sample of training dataset part 11

Document		
Document number	Content	Proposed class
91	As IBM employees, we may face ethical and legal questions	permission
92	As part of IBM's Globally Integrated Enterprise, your workplace may include working from an IBM location, a client location, or your home	permission
93	Information on how to report and protect intellectual property can be found at the Intellectual Property & Licensing site	permission
94	Software may be on tangible media (e.g. CDs, portable devices and publications), or it may be downloadable or accessible for use online	permission
95	Incidental personal use of such property and systems—meaning use that is limited in duration, does not violate company policies, and does not interfere with doing your job—may be permitted by management	permission
96	In addition, in order to protect its employees, assets, and business interests, IBM may share outside of IBM anything it finds, such as with its outside legal or other advisors, or with law enforcement	permission
97	Only the efficient use of all resources at all levels can ensure the Company's success in the long term	permission
98	then those devices may also be examined by IBM	permission

Table 12: Annotated document sample of training dataset part 12

Document		
Document number	Content	Proposed class
99	An IBM Business Partner may be both a client and a competitor	permission
100	Another organization may be an IBM supplier and client at the same time	permission
101	Even appearing to do so can undermine the integrity of our established procedures	permission
102	Many of these contacts are acceptable as long as established procedures are followed	permission
103	Discussion or collaboration on prohibited subjects with competitors can be illegal	permission
104	As part of your work, you may have access to personal information, such as information about consumers or employees of clients, suppliers, IBM Business Partners and others	permission
105	You may only use such information to the extent necessary to fulfill your assigned job responsibilities and in accordance with instructions issued by management or applicable IBM policies, directives, and guidelines	permission
106	The terms, restrictions and other conditions that apply to using confidential information can vary widely so it is important that you understand and comply with the applicable obligations	permission

Table 13: Annotated document sample of training dataset part 13

Document		
Document number	Content	Proposed class
107	Under these guidelines, senior executive management may approve receiving or giving higher value gifts and business amenities provided the gifts and business amenities are not prohibited by law or known client, business partner or supplier practices	permission
108	you may accept the following”	permission
109	Likewise, in your work you may have access to personal information of others	permission
110	Within these processes, authority for pricing, contract terms and conditions and other actions may have been delegated to certain functions and to line management	permission

Table 14: Annotated document sample of training dataset part 14

Document		
Document number	Content	Proposed class
111	Certain off-the-job activities can affect your IBM position, or can otherwise reflect negatively on IBM	permission
112	Each individual employee can contribute to constantly improving the quality of our products and help us keep our product promises	permission
113	The Compliance Officer can also help you	permission
114	Each individual employee can contribute to ensuring a safe working environment at Beiersdorf	permission
115	Each individual employee can contribute to Beiersdorf's success by respecting the Company's diversity	prohibition
116	With respect to gifts and invitations, too, only absolute transparency can avoid damage to Beiersdorf and the employees concerned	permission
117	Low-value presents may only be granted in exceptional cases provided that they are appropriate and no consideration is expected	permission
118	Gifts and invitations may give the impression that the person making the gift or the invitation expects benefits as a result	permission
119	Further information on dealing with invitations and gifts as well as guidance on what is “appropriate” can be found in Beiersdorf's anticorruption guidelines	permission
120	Confidential information may only be used for business purposes	permission

Table 15: Annotated document sample of training dataset part 15

Appendix E

In this appendix, the confusion matrix for SVC family models trained in all 10 rounds of executing the prediction model on our data is presented.

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**9**	8	0	0	17
	Obligation	2	**55**	1	10	68
	Prohibition	0	10	**10**	0	20
	Permission	0	8	2	**2**	12
	Total	11	73	13	12	109

Table 16: SVC Linear 1st round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**15**	2	0	0	17
	Obligation	1	**66**	0	0	67
	Prohibition	0	3	**17**	0	20
	Permission	0	9	1	**2**	12
	Total	16	80	18	2	116

Table 17: SVC Linear 2nd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**12**	4	1	0	17
	Obligation	2	**61**	4	0	67
	Prohibition	0	2	**18**	0	20
	Permission	0	9	0	**2**	11
	Total	14	76	23	2	115

Table 18: SVC Linear 3rd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**12**	4	1	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	7	**13**	0	20
	Permission	0	6	1	**4**	11
	Total	12	84	15	4	115

Table 19: SVC Linear 4th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**12**	5	0	0	17
	Obligation	0	**65**	2	0	67
	Prohibition	0	2	**18**	0	20
	Permission	0	6	0	**5**	11
	Total	12	78	20	5	115

Table 20: SVC Linear 5th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**13**	3	1	0	17
	Obligation	1	**64**	2	0	67
	Prohibition	0	2	**16**	1	19
	Permission	0	6	0	**5**	11
	Total	14	75	19	6	114

Table 21: SVC Linear 6th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**14**	3	0	0	17
	Obligation	0	**65**	1	1	67
	Prohibition	0	1	**17**	1	19
	Permission	0	7	0	**4**	11
	Total	14	76	18	6	114

Table 22: SVC Linear classifier 7th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**14**	3	0	0	17
	Obligation	0	**66**	1	0	67
	Prohibition	0	4	**15**	0	19
	Permission	0	6	1	**4**	11
	Total	14	79	17	4	114

Table 23: SVC Linear classifier 8th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**11**	4	2	0	17
	Obligation	0	**66**	1	0	67
	Prohibition	0	6	**13**	0	19
	Permission	0	7	0	**4**	11
	Total	11	83	16	4	114

Table 24: SVC Linear classifier 9th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**12**	3	1	0	16
	Obligation	1	**66**	0	0	67
	Prohibition	0	7	**12**	0	19
	Permission	0	10	0	**1**	11
	Total	13	86	13	1	113

Table 25: SVC Linear classifier 10th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**68**	0	0	68
	Prohibition	0	20	**0**	0	20
	Permission	0	12	0	**0**	12
	Total	0	117	0	0	117

Table 26: SVC Linear with $c = 0.025$ 1st round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	20	**0**	0	20
	Permission	0	12	0	**0**	12
	Total	0	116	0	0	116

Table 27: SVC Linear with $c = 0.025$ 2nd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	20	**0**	0	20
	Permission	0	11	0	**0**	11
	Total	0	115	0	0	115

Table 28: SVC Linear with $c = 0.025$ 3rd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	20	**0**	0	20
	Permission	0	11	0	**0**	11
	Total	0	115	0	0	115

Table 29: SVC Linear with $c = 0.025$ 4th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	20	**0**	0	20
	Permission	0	11	0	**0**	11
	Total	0	115	0	0	115

Table 30: SVC Linear with $c = 0.025$ 5th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**0**	0	19
	Permission	0	11	0	**0**	11
	Total	0	114	0	0	114

Table 31: SVC Linear with $c = 0.025$ 6th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**0**	0	19
	Permission	0	11	0	**0**	11
	Total	0	114	0	0	114

Table 32: SVC Linear with $c = 0.025$ 7th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**0**	0	19
	Permission	0	11	0	**0**	11
	Total	0	114	0	0	114

Table 33: SVC Linear with $c = 0.025$ 8th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**0**	0	19
	Permission	0	11	0	**0**	11
	Total	0	114	0	0	114

Table 34: SVC Linear with $c = 0.025$ 9th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	16	0	0	16
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**0**	0	19
	Permission	0	11	0	**0**	11
	Total	0	113	0	0	113

Table 35: SVC Linear with $c = 0.025$ 10th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**1**	16	0	0	17
	Obligation	0	**68**	0	0	68
	Prohibition	0	20	**0**	0	20
	Permission	0	12	0	**0**	12
	Total	0	117	0	0	117

Table 36: SVC with $c = 1$ and $gamma = 2$ 1st round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**5**	12	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	20	**0**	0	20
	Permission	0	12	0	**0**	12
	Total	0	116	0	0	116

Table 37: SVC with $c = 1$ and $gamma = 2$ 2nd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**3**	14	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**1**	0	20
	Permission	0	11	0	**0**	11
	Total	3	111	1	0	115

Table 38: SVC with $c = 1$ and $gamma = 2$ 3rd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**1**	16	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	18	**2**	0	20
	Permission	0	11	0	**0**	11
	Total	1	112	2	0	115

Table 39: SVC with $c = 1$ and $gamma = 2$ 4th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**6**	11	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	20	**0**	0	20
	Permission	0	11	0	**0**	11
	Total	6	109	0	0	115

Table 40: SVC with $c = 1$ and $gamma = 2$ 5th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**0**	0	19
	Permission	0	11	0	**0**	11
	Total	0	114	0	0	114

Table 41: SVC with $c = 1$ and $gamma = 2$ 6th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**2**	15	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**0**	0	19
	Permission	0	11	0	**0**	11
	Total	2	112	0	0	114

Table 42: SVC with $c = 1$ and $gamma = 2$ 7th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	18	**1**	0	19
	Permission	0	11	0	**0**	11
	Total	0	113	1	0	114

Table 43: SVC with $c = 1$ and $gamma = 2$ 8th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**1**	16	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**0**	0	19
	Permission	0	11	0	**0**	11
	Total	1	113	0	0	114

Table 44: SVC with $c = 1$ and $gamma = 2$ 9th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	16	0	0	16
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**0**	0	19
	Permission	0	11	0	**0**	11
	Total	0	113	0	0	113

Table 45: SVC with $c = 1$ and $gamma = 2$ 10th round confusion matrix for 4 classes

Appendix F

In this appendix, the confusion matrix for Adaboost classifier models trained in all 10 rounds of executing the prediction model on our data is presented.

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**13**	4	0	0	17
	Obligation	0	**66**	2	0	68
	Prohibition	0	2	**18**	0	20
	Permission	0	10	1	**1**	12
	Total	13	82	21	1	117

Table 46: Adaboost classifier 1st round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**10**	7	0	0	17
	Obligation	5	**61**	0	1	67
	Prohibition	1	3	**15**	1	20
	Permission	1	4	1	**6**	12
	Total	17	75	16	8	116

Table 47: Adaboost classifier 2nd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**12**	1	3	1	17
	Obligation	3	**49**	2	13	67
	Prohibition	0	6	**14**	0	20
	Permission	0	6	1	**4**	11
	Total	15	62	20	18	115

Table 48: Adaboost classifier 3rd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**14**	3	0	0	17
	Obligation	3	**61**	1	2	67
	Prohibition	0	6	**13**	1	20
	Permission	0	3	1	**7**	11
	Total	17	73	15	10	115

Table 49: Adaboost classifier 4th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**11**	6	0	0	17
	Obligation	1	**59**	3	4	67
	Prohibition	3	10	**7**	0	20
	Permission	0	8	0	**3**	11
	Total	15	83	10	7	115

Table 50: Adaboost classifier 5th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**12**	5	0	0	17
	Obligation	3	**59**	0	5	67
	Prohibition	1	7	**11**	0	19
	Permission	0	8	1	**2**	11
	Total	16	79	12	7	114

Table 51: Adaboost classifier 6th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**14**	3	0	0	17
	Obligation	0	**58**	2	7	67
	Prohibition	0	7	**12**	0	19
	Permission	1	7	0	**3**	11
	Total	15	75	14	10	114

Table 52: Adaboost classifier 7th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**14**	1	2	0	17
	Obligation	5	**50**	1	11	67
	Prohibition	0	7	**11**	1	19
	Permission	0	7	1	**3**	11
	Total	19	65	15	15	114

Table 53: Adaboost classifier 8th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**9**	8	0	0	17
	Obligation	1	**55**	2	9	67
	Prohibition	0	4	**14**	1	19
	Permission	0	9	0	**2**	11
	Total	10	76	16	12	114

Table 54: Adaboost classifier 9th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**13**	3	0	0	16
	Obligation	3	**49**	4	11	67
	Prohibition	0	8	**10**	1	19
	Permission	0	3	1	**7**	11
	Total	16	63	15	19	113

Table 55: Adaboost classifier 10th round confusion matrix for 4 classes

Appendix G

In this appendix, the confusion matrix for Random Forest classifier models trained in all 10 rounds of executing the prediction model on our data is presented.

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**12**	5	0	0	17
	Obligation	0	**66**	2	0	68
	Prohibition	1	8	**11**	0	20
	Permission	1	7	2	**2**	12
	Total	14	86	15	2	117

Table 56: Random Forest classifier 1st round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**12**	4	1	0	17
	Obligation	2	**61**	2	2	67
	Prohibition	1	3	**16**	0	20
	Permission	0	5	0	**7**	12
	Total	15	73	19	9	116

Table 57: Random Forest classifier 2nd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**16**	1	0	0	17
	Obligation	5	**58**	3	1	67
	Prohibition	1	5	**14**	0	20
	Permission	2	6	0	**3**	11
	Total	24	70	17	4	115

Table 58: Random Forest classifier 3rd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**13**	3	1	0	17
	Obligation	6	**59**	1	1	67
	Prohibition	0	12	**8**	0	20
	Permission	0	6	0	**5**	11
	Total	19	80	10	6	115

Table 59: Random Forest classifier 4th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**12**	5	0	0	17
	Obligation	1	**64**	2	0	67
	Prohibition	2	4	**14**	0	20
	Permission	0	7	0	**4**	11
	Total	15	80	16	4	115

Table 60: Random Forest classifier 5th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**12**	3	1	1	17
	Obligation	5	**60**	0	2	67
	Prohibition	0	4	**14**	1	19
	Permission	0	4	0	**7**	11
	Total	17	71	15	11	114

Table 61: Random Forest classifier 6th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**14**	3	0	0	17
	Obligation	6	**61**	0	0	67
	Prohibition	1	3	**14**	1	19
	Permission	2	7	0	**2**	11
	Total	23	74	14	3	114

Table 62: Random Forest classifier 7th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**14**	3	0	0	17
	Obligation	3	**61**	2	1	67
	Prohibition	3	3	**13**	0	19
	Permission	0	5	1	**5**	11
	Total	20	72	16	6	114

Table 63: Random Forest classifier 8th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**11**	6	0	0	17
	Obligation	1	**65**	1	0	67
	Prohibition	1	5	**13**	0	19
	Permission	0	8	8	**3**	11
	Total	13	84	14	3	114

Table 64: Random Forest classifier 9th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**13**	1	1	1	16
	Obligation	1	**66**	0	0	67
	Prohibition	0	13	**6**	0	19
	Permission	0	9	0	**2**	11
	Total	14	89	7	3	113

Table 65: Random Forest classifier 10th round confusion matrix for 4 classes

Appendix H

In this appendix, the confusion matrix for Naive Bayes family classifier models trained in all 10 rounds of executing the prediction model on our data is presented.

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**10**	6	1	0	17
	Obligation	1	**63**	4	0	68
	Prohibition	0	9	**10**	1	20
	Permission	1	8	0	**3**	12
	Total	12	86	15	4	117

Table 66: Gaussian Naive Bayes classifier 1st round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**10**	6	1	0	17
	Obligation	0	**64**	21	1	67
	Prohibition	0	6	**13**	1	20
	Permission	0	7	3	**2**	12
	Total	10	83	19	4	116

Table 67: Gaussian Naive Bayes classifier 2nd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**5**	7	5	0	17
	Obligation	0	**61**	3	3	67
	Prohibition	0	10	**10**	0	20
	Permission	0	8	1	**2**	11
	Total	5	86	19	5	115

Table 68: Gaussian Naive Bayes classifier 3rd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**4**	10	2	1	17
	Obligation	0	**64**	2	1	67
	Prohibition	0	9	**11**	0	20
	Permission	0	8	1	**2**	11
	Total	4	91	16	4	115

Table 69: Gaussian Naive Bayes classifier 4th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**10**	6	0	1	17
	Obligation	0	**63**	3	1	67
	Prohibition	0	8	**11**	1	20
	Permission	0	5	4	**2**	11
	Total	10	82	18	5	115

Table 70: Gaussian Naive Bayes classifier 5th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**8**	8	0	1	17
	Obligation	0	**64**	3	0	67
	Prohibition	0	7	**11**	1	19
	Permission	0	11	0	**0**	11
	Total	8	90	14	2	114

Table 71: Gaussian Naive Bayes classifier 6th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**5**	12	0	0	17
	Obligation	1	**61**	2	3	67
	Prohibition	0	5	**14**	0	19
	Permission	0	7	0	**4**	11
	Total	6	85	16	7	114

Table 72: Gaussian Naive Bayes classifier 7th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**10**	5	2	0	17
	Obligation	0	**62**	3	2	67
	Prohibition	0	8	**11**	0	19
	Permission	0	7	2	**2**	11
	Total	10	82	18	4	114

Table 73: Gaussian Naive Bayes classifier 8th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**7**	6	3	1	17
	Obligation	0	**66**	1	0	67
	Prohibition	0	8	**11**	0	19
	Permission	0	7	0	**4**	11
	Total	7	87	15	5	114

Table 74: Gaussian Naive Bayes classifier 9th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**4**	9	3	0	16
	Obligation	0	**65**	1	1	67
	Prohibition	0	8	**11**	0	19
	Permission	0	7	1	**3**	11
	Total	4	89	16	4	113

Table 75: Gaussian Naive Bayes classifier 10th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**12**	5	0	0	17
	Obligation	0	**67**	1	0	68
	Prohibition	0	19	**1**	0	20
	Permission	0	12	0	**0**	12
	Total	2	113	2	0	117

Table 76: Multinomial Naive Bayes classifier 1st round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**4**	13	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**1**	0	20
	Permission	0	12	0	**0**	12
	Total	4	111	1	0	116

Table 77: Multinomial Naive Bayes classifier 2nd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**3**	14	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**1**	0	20
	Permission	0	11	0	**0**	11
	Total	3	111	1	0	115

Table 78: Multinomial Naive Bayes classifier 3rd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**1**	16	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	18	**2**	0	20
	Permission	0	11	0	**0**	11
	Total	1	112	2	0	115

Table 79: Multinomial Naive Bayes classifier 4th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**4**	13	0	0	17
	Obligation	0	**66**	1	0	67
	Prohibition	0	20	**0**	0	20
	Permission	0	11	0	**0**	11
	Total	4	110	1	0	115

Table 80: Multinomial Naive Bayes classifier 5th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**1**	16	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**0**	0	19
	Permission	0	11	0	**0**	11
	Total	1	113	0	0	114

Table 81: Multinomial Naive Bayes classifier 6th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	17	**2**	0	19
	Permission	0	11	0	**0**	11
	Total	0	112	2	0	114

Table 82: Multinomial Naive Bayes classifier 7th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**1**	16	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	17	**2**	0	19
	Permission	0	11	0	**0**	11
	Total	1	111	2	0	114

Table 83: Multinomial Naive Bayes classifier 8th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**1**	16	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	18	**1**	0	19
	Permission	0	11	0	**0**	11
	Total	1	112	1	0	114

Table 84: Multinomial Naive Bayes classifier 9th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**2**	14	0	0	16
	Obligation	0	**67**	0	0	67
	Prohibition	0	17	**2**	0	19
	Permission	0	11	0	**0**	11
	Total	2	109	2	0	113

Table 85: Multinomial Naive Bayes classifier 10th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**68**	0	0	68
	Prohibition	0	20	**0**	0	20
	Permission	0	12	0	**0**	12
	Total	0	117	0	0	117

Table 86: Bernoulli Naive Bayes classifier 1st round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	20	**0**	0	20
	Permission	0	12	0	**0**	12
	Total	0	116	0	0	116

Table 87: Bernoulli Naive Bayes classifier 2nd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	20	**0**	0	20
	Permission	0	11	0	**0**	11
	Total	0	115	0	0	115

Table 88: Bernoulli Naive Bayes classifier 3rd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	20	**0**	0	20
	Permission	0	11	0	**0**	11
	Total	0	115	0	0	115

Table 89: Bernoulli Naive Bayes classifier 4th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	20	**0**	0	20
	Permission	0	11	0	**0**	11
	Total	0	115	0	0	115

Table 90: Bernoulli Naive Bayes classifier 5th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**0**	0	19
	Permission	0	11	0	**0**	11
	Total	0	114	0	0	114

Table 91: Bernoulli Naive Bayes classifier 6th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**0**	0	19
	Permission	0	11	0	**0**	11
	Total	0	114	0	0	114

Table 92: Bernoulli Naive Bayes classifier 7th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**0**	0	19
	Permission	0	11	0	**0**	11
	Total	0	114	0	0	114

Table 93: Bernoulli Naive Bayes classifier 8th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	17	0	0	17
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**0**	0	19
	Permission	0	11	0	**0**	11
	Total	0	114	0	0	114

Table 94: Bernoulli Naive Bayes classifier 9th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**0**	16	0	0	16
	Obligation	0	**67**	0	0	67
	Prohibition	0	19	**0**	0	19
	Permission	0	11	0	**0**	11
	Total	0	113	0	0	113

Table 95: Bernoulli Naive Bayes classifier 10th round confusion matrix for 4 classes

Appendix I

In this appendix, the confusion matrix for Multilayer Perceptron classifier trained in all 10 rounds of executing the prediction model on our data is presented.

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**15**	1	1	0	17
	Obligation	0	**64**	4	0	68
	Prohibition	0	1	**19**	0	20
	Permission	0	8	3	**1**	12
	Total	15	74	27	1	117

Table 96: Multilayer perceptron classifier 1st round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**15**	2	0	0	17
	Obligation	1	**65**	1	0	67
	Prohibition	0	5	**15**	0	20
	Permission	0	8	3	**1**	12
	Total	16	80	19	1	116

Table 97: Multilayer perceptron classifier 2nd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**15**	1	1	0	17
	Obligation	3	**59**	5	0	67
	Prohibition	0	0	**20**	0	20
	Permission	0	7	2	**2**	11
	Total	18	67	28	2	115

Table 98: Multilayer perceptron classifier 3rd round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**14**	2	1	0	17
	Obligation	2	**62**	3	0	67
	Prohibition	0	6	**14**	0	20
	Permission	0	4	2	**5**	11
	Total	16	74	20	5	115

Table 99: Multilayer perceptron classifier 4th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**13**	4	0	0	17
	Obligation	0	**64**	3	0	67
	Prohibition	1	2	**17**	0	20
	Permission	0	4	0	**7**	11
	Total	14	74	20	7	115

Table 100: Multilayer perceptron classifier 5th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**14**	1	1	1	17
	Obligation	1	**64**	2	0	67
	Prohibition	0	2	**16**	1	19
	Permission	0	6	1	**4**	11
	Total	15	73	20	6	114

Table 101: Multilayer perceptron classifier 6th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**14**	2	1	0	17
	Obligation	3	**61**	2	1	67
	Prohibition	0	1	**17**	1	19
	Permission	1	3	1	**6**	11
	Total	18	67	21	8	114

Table 102: Multilayer perceptron 7th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**16**	1	0	0	17
	Obligation	1	**62**	3	1	67
	Prohibition	0	3	**16**	0	19
	Permission	0	3	2	**6**	11
	Total	17	69	21	7	114

Table 103: Multilayer perceptron 8th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**13**	2	2	0	17
	Obligation	0	**64**	3	0	67
	Prohibition	0	5	**14**	0	19
	Permission	2	2	0	**7**	11
	Total	15	73	19	7	114

Table 104: Multilayer perceptron 9th round confusion matrix for 4 classes

		Predicted				
		Condition	Obligation	Prohibition	Permission	Total
Actual	Condition	**11**	4	1	0	16
	Obligation	1	**66**	1	0	67
	Prohibition	0	8	**11**	0	19
	Permission	2	7	1	**3**	11
	Total	12	85	13	3	113

Table 105: Multilayer perceptron 10th round confusion matrix for 4 classes