TEXT MINING:
A BURGEONING QUALITY IMPROVEMENT TOOL

MOHAMMAD ALKIN MOHAMMAD

NOVEMBER 2007

TEXT MINING:
A BURGEONING QUALITY IMPROVEMENT TOOL

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

MOHAMMAD ALKIN MOHAMMAD

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF SCIENTIFIC COMPUTING

NOVEMBER 2007

Approval of the Graduate School of Applied Mathematics

_____

Prof. Dr. Ersan Akyıldız

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____

Prof. Dr. Bülent Karasözen

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Prof. Dr. Gerhard Wilhelm Weber

Supervisor

Examining Committee Members

Prof. Dr. Gerhard Wilhelm Weber	(METU, IAM) _____
Prof. Dr. Gülser Köksal	(METU, IE) _____
Prof. Dr. Bülent Karasözen	(METU, IAM) _____
Assist. Prof. Dr. Hakan Öktem	(METU, IAM) _____
Dr. Ömür Uğur	(METU, IAM) _____

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

Name, Last name: Muhammed Alkın Muhammed

Signature:

ABSTRACT

TEXT MINING:
A BURGEONING QUALITY IMPROVEMENT TOOL

J. Mohammad, Mohammad Alkin
M.Sc., Department of Scientific Computing
Supervisor: Prof. Dr. Gerhard Wilhelm Weber

November 2007, 105 pages

While the amount of textual data available to us is constantly increasing, managing the texts by human effort is clearly inadequate for the volume and complexity of the information involved. Consequently, requirement for automated extraction of useful knowledge from huge amounts of textual data to assist human analysis is apparent. Text mining (*TM*) is mostly an automated technique that aims to discover knowledge from textual data. In this thesis, the notion of text mining, its techniques, applications are presented. In particular, the study provides the definition and overview of concepts in *text categorization*. This would include *document representation* models, *weighting schemes*, *feature selection* methods, *feature extraction*, *performance measure* and machine learning techniques. The thesis details the functionality of text mining as a quality improvement tool. It carries out an extensive survey of text mining applications within service sector and manufacturing industry. It presents two broad experimental studies tackling the potential use of text mining for the hotel industry (the comment card analysis), and in automobile manufacturer (miles per gallon analysis).

Keywords: Text Mining, Text Categorization, Quality Improvement, Service Sector, Manufacturing Industry.

ÖZ

METİN VERİ MADENCİLİĞİ:
HIZLA GELİŞEN BİR KALİTE İYİLEŞTİRME ARACIDIR

J. Mohammad, Mohammad Alkin
Yüksek Lisans, Bilimsel Hesaplama Bölümü
Tez Yöneticisi: Prof. Dr. Gerhard Wilhelm Weber

Kasım 2007, 105 sayfa

Elimizdeki metin verileri sürekli artmakla beraber, içerdiği bilginin karmaşalılığı ve hacmi nedeniyle metinleri, insan gayretiyle yönetmek kesinlikle yetersiz kalmaktadır. Netice itibariyle, insan analizine yardımcı olabilmesi için, otomatikleştirilen çok büyük hacimlerdeki metin verilerinden yararlı bilgileri çıkarma gerksinimi apaçıktır. Metin veri madenciliği (*MVM*), çoğunlukla, metin verilerinden bilgi bulmayı hedefleyen otomatikleştirilmiş bir yöntemdir. Bu tezde, metin veri madenciliğinin kavramı, teknikleri ve uygulamaları sunulmaktadır. Çalışma, özellikle, metin sınıflandırmayla ilgili kavramların tanıtım ve ana fikirlerini sağlamaktadır. Bu da, doküman simgeleme modelleri, ağırlık verme metotları, özellik seçme metotları, özellik çıkarma, performans değerlendirme ve makine öğrenim tekniklerini içermektedir. Tez, metin veri madenciliğin bir kalite iyileştirme aracı olma görevselliğinden ayrıntılı bahsetmektedir. Hizmet sektörü ve imalat endüstrisinde metin veri madenciliği uygulamaları ile ilgili kapsamlı incelemeler yapılmaktadır. Tez, otelcilik sektörü (yorum kartı analizi) ve otomobil imalatı (galonda kaç mil) üzerinde uygulanan metin veri madenciliğinin iki geniş deneysel çalışmasını sunmaktadır.

Ana Kelimeler: Metin Veri Madenciliği, Metin Sınıflandırma, Kalite İyileştirme, Hizmet Sektörü, İmalat Endüstrisi.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

We are in an age often referred to as the information age. As we believe that information leads to power and success, and owing to complicated technologies such as computers, satellites, etc., in this age, we have been collecting enormous amounts of information. Initially, with the advent of computers and means for mass digital storage, we started collecting and storing all sorts of data, relying upon the power of computers to help sort through this mixture of information [80, 175]. Today, we have far more information than we can handle: from business transactions and scientific data, to satellite pictures, text reports and military intelligence [175]. According to *Moore's law,* the number of transistors in a single microchip is doubled every 18 months, and the growth of the semiconductor industry has so far followed the prediction. We can correlate this with a similar observation from the data and information domain. If the amount of information in the world doubles every 20 months, the size and number of databases probably increases at a similar rapidity [106, 139, 156]. Databases today can range in size into the terabytes (more than 1,000,000,000,000 bytes of data). Within these masses of data lies hidden information of strategic importance [158]. Discovery of knowledge from this huge volume of data is a challenge indeed. *Data mining* (*DM*) is an attempt to make sense of the information explosion embedded in this huge volume of data [106], which is being used both to increase revenues and to reduce costs. The potential returns are enormous. Innovative organizations worldwide are already using data mining to locate and appeal to higher-value customers, to reconfigure their product offerings to increase sales, and to minimize losses due to error or fraud [158]. The primary focus of this thesis is on the application of data mining techniques to databases with textual content, or so called *text mining* (*TM*), to improve the quality of products and/or services.

1.1 Data Mining

Data Mining (*DM*), also popularly known as *Knowledge Discovery in Databases* (*KDD*), refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases. While data mining and *KDD* are frequently treated as synonyms, data mining is actually part of the knowledge discovery process. Figure 1.1 displays data mining as a step in an iterative knowledge discovery process [44, 69, 175].

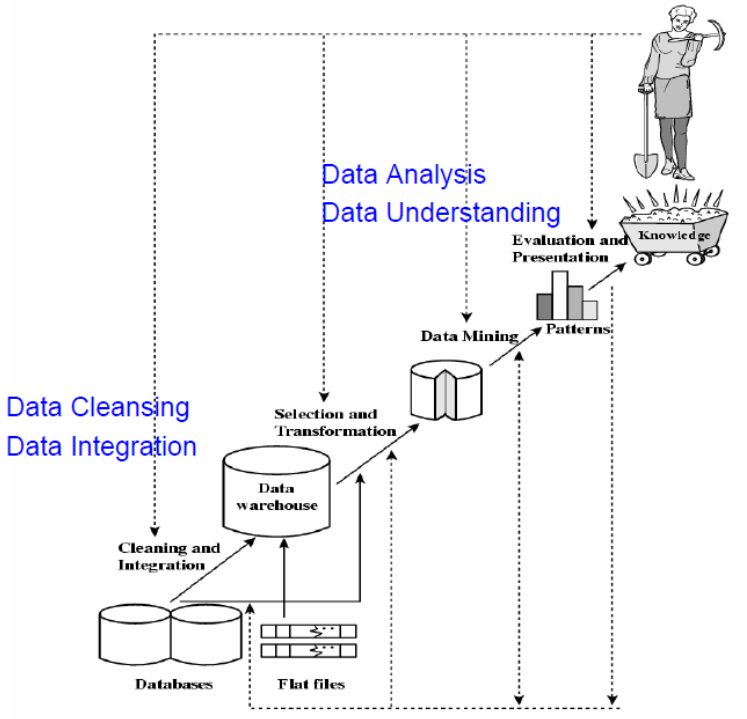

Figure 1.1: *Data mining is an essential step in knowledge discovery* [18].

There are other definitions:

- "Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner" [90].

- "Data mining is an interdisciplinary field bringing together techniques from machine learning, pattern recognition, statistics, databases, and visualization to address the issue of information extraction from large data bases" [44, 90, 110].
- Data mining is the process of discovering meaningful new correlations, patterns and trends by sifting through large amounts of data stored in repositories, using pattern recognition technologies as well as statistical and mathematical techniques [44, 90, 110].

Data mining provides a critical advantage, when utilized, in fields such as engineering, economics and management science. The high volume of activity and unprecedented growth provided a fertile ground for data mining to flourish [119]. According to the online technology magazine *ZDNET News*, it is predicted that data mining will be "one of the most revolutionary developments of the next decade" [90].

1.2 Text Mining

As research in all areas of life continues, many fields will become so overwhelmed with information that processing all the information on a particular topic will become actually impossible for any person. There is a vast amount of unstructured, both documents and Web pages, business information in data repositories on Intranets and the Internet. In fact, it is estimated that 80% of a company's information, such as emails, memos, customer correspondence, and reports is contained in text documents [39, 80, 146, 153]. The ability to distil this untapped source of information, free text document, provides substantial competitive advantages for a company to succeed in the era of a knowledge-based economy [39]. Since managing the texts by human effort, have become both inadequate and too expensive to perform and to maintain for the majority of the available data, the use of automatic methods, algorithms, and tools for dealing with this large amounts of textual data, has become necessary [87]. The *Text Mining* (*TM*) field was born to address the huge demand for mining large amounts of text automatically [43, 145]. It is inherently interdisciplinary, borrowing heavily from neighbouring fields such as data mining and computational linguistics [10], (see Figure 1.2).

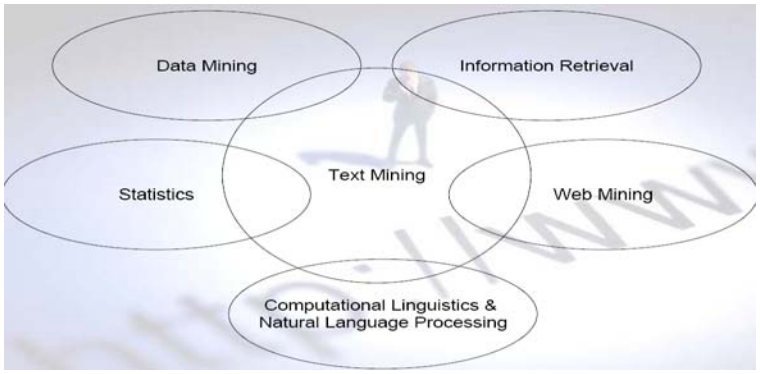

Figure 1.2: TM is an interdisciplinary field [20].

We can define *text mining* as the discovery by computer of new, previously unknown information, by automatically extracting information from a usually large amount of different unstructured textual resources. It can be viewed as an extension of data mining or knowledge discovery from (structured) databases [20, 43].
In any case, a scheme can be built to include all possible steps that characterize text mining strategies. The scheme below (Table 1.1) describes the steps:

Table 1.1: All possible steps that characterize TM strategies [31].

Document Pre-processing	
1	Definition of rules for extraction/collection of text (data selection and filtering)
2	Definition and identification of document format
3	Text normalization (cleaning, recognition of dates and of currencies,…)
4	Reduction and transformation of text (elimination of stop words, identification of proper names)
Lexical Processing	
5	Choice of unit analysis: Words (tokens or lemmas) and multiword expressions, terms
6	Definition of grammatical rules to solve text ambiguity
7	Linguistic and lexical analysis (lemmatisation, key words detection, other tagging)
8	Definition of semantic categories to be searched for in the text (marking of terminology of interest),extraction of key words
9	Classification according to concepts and/or other metadata, information extraction
TM Processing	
10	Classification of texts
11	Clustering of texts and summarisation
12	Knowledge extraction (in some cases integrated with the aid of experts)
13	Visualisation techniques
14	Integration of TM results with data mining processes

Most text mining objectives fall under nine categories of operations: *entity extraction, text-base navigation, search and retrieval, clustering, categorization, summarization, trends analysis, associations, and visualizations,* [142].

In this thesis text categorization is considered in some detail.

Entity extraction deals with finding particular pieces of information within a text [39]. It is to distinguish which noun phrase is a person, place, organization or other distinct objects. This operation should include term extractions and calculate the number of times each term appears in the text analyzed (keyword frequency) [142]. *Text-base navigation* enables the text miner to see related terms in context and connect important relationships between them [142]. *Search and retrieve* operation allows the user to search and retrieve relevant information based on pre-specified search criteria. *Clustering* groups similar documents in a way that the degree of association between two documents is maximal if they belong to the same group and minimal if otherwise [39, 142, 149]. *Categorization* is the process of using content-mining technologies to identify and organize like/similar pieces of raw data into a pre-defined set of topics for analysis [37]. *Summarization* is the operation to describe the content of a document while reducing the amount of text a user must read [39]. *Trends analysis* is used for discovering trends from time-dependent textual data [142]. *Association analysis* is to associate one extracted pattern with another pattern found [142]. *Visualizations* utilize feature extraction and key term indexing in order to build a graphical representation that can help user identifying the main topics or concepts by their importance on the representation. Additionally, it is easy to discover the location of specific documents in a graphical document representation [80, 142].

Text mining techniques can range from simple one (e.g., arithmetic averages) to those with intermediate complexity (e.g., linear regression, clustering and decision trees) and highly complicated ones such as neural network [145].

In the following subsections, the relationship between text mining and data mining, between text mining and natural language processing (*NLP*) and between text mining and information retrieval (*IR*), is discussed.

1.3 Text Mining and Data Mining

Text mining, also known as *text data mining* or *knowledge discovery from textual databases* (*KDTD*), the process of finding useful or interesting patterns, models, directions, trends, or rules from unstructured text, is used to describe the application of data mining techniques to automated discovery of knowledge from text [17, 60, 110]. Text mining has been viewed as natural extension of data mining or sometimes considered as a task of applying the same data mining techniques to the domain of textual information [33, 110]. This reflects the fact that the advent of text mining relies on the burgeoning field of data mining to a great degree, but as the most natural form of storing information is *text*, text mining is believed to have a commercial potential higher than that of data mining [110]. Text mining, however, is also a much more complex task (than data mining) as it involves dealing with text data that are inherently unstructured and fuzzy [153].
Although Text Mining and Data Mining are related as they are mining processes they differ in point of the following issues [144]:

1. Text mining deals with unstructured or semi-structured data, such as text found in articles, documents, etc. However Data Mining is related to structured data from large databases. In addition, another characteristic of text mining is the amount of textual data. The concepts constrained in a text are usually rather abstract and can hardly be modelled by using conventional knowledge representation structures.
2. Furthermore, the occurrence of synonyms (different words with the same meaning) or homonyms (words with the same spelling but with distinct meanings) makes it difficult to detect valid relationships between different parts of the text.

1.4 Text mining and Natural Language Processing

Natural language processing (*NLP*) (Figure 1.3) is the study of computer processing of human language [71]. It is the ability to automatically process written text based

on language constructs (words, phrases, sentences, etc.) and different parts of speech (nouns, adjectives, verbs, etc.) [37]. *NLP* has developed various techniques that are typically linguistically inspired, i.e., text is typically syntactically parsed using information from a formal grammar and a lexicon, the resulting information is then interpreted semantically and used to extract information about what was said. *NLP* may be deep (parsing every part of every sentence and attempting to account semantically for every part) or shallow (parsing only certain passages or phrases within sentences or producing only limited semantic analysis), and may even use statistical means to disambiguate word senses or multiple parses of the same sentence. It tends to focus on one document or piece of text at a time and be rather computationally expensive. It includes techniques like word stemming (removing suffixes) or a related technique, lemmatization (replacing an inflected word with its base form), part-of-speech (*POS*) tagging (elaborations on noun, verb, preposition, etc.) [79]. Text mining appears to comprise the whole of automatic natural language processing and, perhaps, far more besides, for example, analysis of linkage structures such as citations in the academic literature and hyperlinks in the Web literature, both useful sources of information that lie outside the traditional domain of natural language processing. But, in fact, most text mining efforts deliberately avoid the deeper aspects of classic natural language processing in favor of shallower techniques more similar to those used in practical information retrieval [167]. Text mining uses techniques primarily developed in the fields of information retrieval, statistics, and machine learning. Its aim typically is not to understand all or even a large part of what a given speaker/writer has said, but rather to extract patterns across a large number of documents. The simplest form of text mining could be considered information retrieval, what typical search engines do. However, more properly text mining consists of areas such as automatic text classification according to some fixed set of categories, text clustering, automatic summarization. While information retrieval and other forms of text mining frequently make use of word stemming, more sophisticated techniques from *NLP* have been rarely used [79].

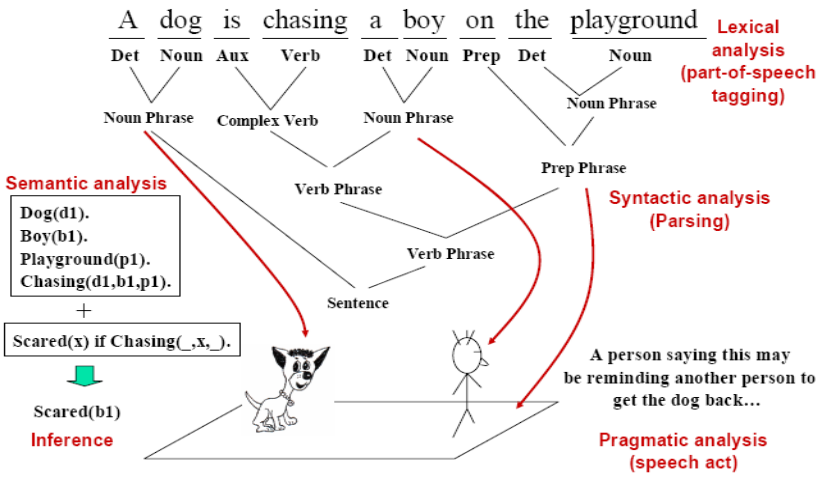

Figure 1.3: Natural Language Processing [62].

1.5 Text Mining and Information Retrieval

It is important to differentiate between text mining and information access (or information retrieval (*IR*), as it is more widely known) [67].

Information retrieval is the finding of documents which contain answers to questions and not the finding of answers itself. In order to achieve this goal statistical measures and methods are used for the automatic processing of text data and comparison to the given question [69]. Even though, the definition of information retrieval is based on the idea of questions and answers, systems that retrieve documents based on keywords, i.e., systems that perform *document retrieval* like most search engines, are frequently also called information retrieval systems [69]. In Information retrieval procedure, the problem is not that the desired information is not known, but rather that it coexists with many other valid pieces of information [20, 67].

While the goal of text mining is to discover or derive new information from data, an information retrieval system can return a document that contains the information a user requested implies that no genuinely new information is found, i.e., no new discovery is being made: the information had to have already been known to the author of the text; otherwise the author could not have written it down [20, 67].

1.6 Rationale

The motivation for the research efforts undertaken in this thesis would be outlined below:

- The huge amounts of textual data and wealth of information within them.
- There have generally been lacks of knowledge of how to handle textual data, and experience in doing that.
- Requirements for automated or semi-automated text analysis schemes.
- Lack of attention paid to textual data within the manufacturing industry and the service sector.
- Comparatively less work has been done to demonstrate how to employ machine learning methods on textual databases.

These concerns give rise to the focus of this thesis.

1.7 Research Efforts

Given the focus, the research efforts undertaken in this thesis could be outlined as follows:

- Looking for real-life textual data pertaining to the service sector and manufacturing industry.
- Trying a wide variety of softwares in looking for an appropriate and available one.
- Applying various text mining operations to textual data from the hotel industry, and automobile manufacturer.
- Providing the algorithms for numerous methods throughout the thesis.
- Organizing the most popular and successful machine learning methods used in text categorization, which are quite general and applicable to other text classification tasks.
- Organizing several text preprocessing approaches which are essential to get effective textual data analysis results.

1.8 Organization of the Thesis

The thesis is organized as given below:

Chapter 1 defines data mining and introduces the notion of text mining. A rough framework for text mining and its main operations are presented. In particular, the relationship between text mining and data mining, between text mining and *NLP*, and between text mining and information retrieval, is discussed.

Chapter 2 provides the definition and overview of concepts in text categorization. This would include text preprocessing, document representation models, weighting schemes, feature selection methods, feature extraction, performance measures and machine learning techniques.

Chapter 3 details the functionality of text mining as a quality improvement tool. It carries out an extensive survey of text mining applications within service sector and manufacturing industry. It presents two broad experimental studies tackling the potential use of text mining for the hotel industry (the comment card analysis), and in automobile manufacturer (miles per gallon analysis). Moreover, further techniques and methods used in the experimental studies are provided.

Chapter 4 presents conclusions drawn on what have been investigated and discovered in this thesis and suggests some directions for future work. It also presents the summary of the research.

CHAPTER 2

TEXT CATEGORIZATION: THE STATE OF THE ART A LITERATURE SURVEY

One of the successful and directly applicable paradigms for helping users in making good and quick selection of textual information of interest is by classifying the different documents according to their topics. The main text classification tasks, that are usually considered distinct, are information retrieval, text categorization, information filtering, and document clustering. However, boundaries between them are not sharp, as all involve grouping of documents based on their contents. Even though most machine learning methods in this research have been developed for text categorization, they are quite general and applicable to other text classification tasks that are usually based on some similarity (or distance) measures between documents [92]. *Text categorization* (Figure 2.2, displays an example of it) is, simply, the task of automatically assigning arbitrary documents to predefined categories (topics or classes) based on their content (see Figure 2.1). There are two different approaches to text categorization. One approach assigns each document to a single category, the one it appears to fit best. And, the other approach is to allow each document to be categorized into every category that it matches well [70, 92, 172].

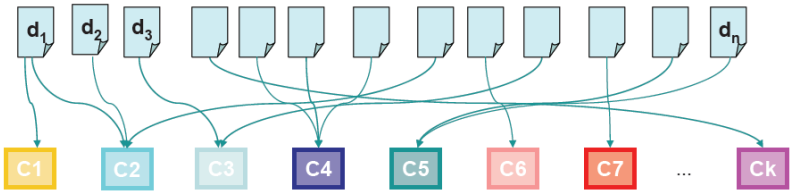

Figure 2.1: Category(ies) assigning [54].

The construction of a text classifier usually involves (i) a phase of *term selection*, in which the most relevant terms for the classification task are identified, (ii) a phase of *term weighting*, in which document weights for the selected terms are computed, and (iii) a phase of *classifier learning*, in which a classifier is generated from the weighted representations of the training documents [31].

Figure 2.2: Automatic text categorization example [16].

2.1 Classification Methods

A large number of approaches has been considered for classification. Among these, the traditional statistical classifiers, the *k*-Nearest Neighbour (*k-NN*) and Naïve Bayes (*NB*), are considered to have the best performance/ complexity ratio [128]. In particular, they were shown to be quite robust on highly dimensional representations [121]. Support Vector Machines (*SVMs*) are another classification method. One of the most important differences of this technique with respect to others is that it only considers a selection of the closest vectors, the so called *support vectors* [128]. Some researchers have shown that *SVMs* has good performance on large data sets [7, 123, 150]. They found that the *SVMs* algorithm outperformed the Naïve Bayes algorithm [7]. Among the vast amount of other possible approaches to text classification, one

can cite decision trees (*ID3, C4.5, CART*), decision rules, neural networks, Bayesian networks, genetic algorithms, and example based classifiers [128].

In the following subsections, three of the classification methods studied in this thesis, namely; 1) Naïve Bayes Classifier, 2) *k*-Nearest Neighbour and 3) Support Vector Machines, are detailed. As can be seen from the surveyed literature thus far, many of the studies carried out have focused on applying these methods on quantitative databases. Comparatively less work has been done to demonstrate how to employ the above-mentioned methods on textual databases although the potential for such data is large and the need for it is immediate. A great deal of effort has been made to do so.

2.1.1 Naïve Bayes Classifier

The basic idea of the Naïve Bayes approach is to use the joint probabilities of words and categories to estimate the probabilities of categories given a document. This approach is based on the Naïve Bayes Theorem, which can be expressed as:

$$P(c_j \mid d) = \frac{P(c_j)P(d \mid c_j)}{P(d)} \tag{2.1}$$

where,

$P(c_j \mid d)$ is the posterior probability of observing class c_j given document vector, d,

$P(d \mid c_j)$ is the prior probability of observing document vector d given occurrence of class c_j,

$P(c_j)$ is the proportion of training examples in category c_j [22],

$P(d)$ is the probability that a randomly picked document has vector d as its representation, [103, 150].

As $P(d)$ is constant for all classes, only $P(d \mid c_j)$ need be computed.

Given data sets with many attributes, it would be extremely computationally expensive to compute $P(d \mid c_j)$.

In order to alleviate this problem it is common to make the assumption that any two coordinates of the document vector are, when viewed as random variables,

statistically independent of each other; this *independence assumption* is encoded by the equation [106, 138]:

$$P(d \mid c_j) = \prod_{k=1}^{m} P(w_k \mid c_j), \tag{2.2}$$

where m is the number of distinct words, w, in the document collection [61]. Probabilistic classifiers that use this assumption are called *Naïve Bayes classifiers*, and account for most of the probabilistic approaches to text categorization in the literature [138].

Because training consists of counting the features in training documents, training time scales linearly with the size of all of the training documents. Classification can be done by looking up the pre-computed probabilities for each of a document's features, so classification scales linearly with the number of categories [150].

The Naïve Bayes algorithm classifies text by, firstly, computing the probability of each term to occur in documents of specific classes in the training set. It then combines the probabilities associated with words found in the document to classify to estimate the probability that this document belongs to different classes. Finally, it assigns the document to the class with the highest probability [127].

The Algorithm

Basic assumption: all terms distribute in documents independently.

1. Input: new document *d.*

2. predefined categories: $C = \{c_1, c_2, ..., c_n\}$.

3. Compute the probability that *d* is in each class $c_j \in C$:

$$P(c_j \mid d) = \frac{P(d \mid c_j)P(c_j)}{P(d)}.$$

As $P(d)$ is constant for all classes, it can be left out. The Naïve Bayes classifier makes an assumption of word independence in order to estimate $P(d \mid c_j)$ as:

$$P(d \mid c_j) = \prod_{k=1}^{m} P(w_k \mid c_j),$$

where m is the number of distinct words, w, in the document collection.

4. Output:

Assigns to *d* the category c_j with the highest probability:

$$P(d \mid c_j) = \max_{c_j \in C}(P(d \mid c_j)).$$

[22, 85]

2.1.2 *K*-Nearest Neighbour

When trying to solve new problems, people often look at solutions to similar problems that they have previously solved. *K-Nearest Neighbour* (*k-NN*) is a classification technique that uses a version of this same method. It decides in which class to place a new case by examining some number (the "*k*" in *k-NN*) of the most similar cases or neighbours (Figure 2.3). It counts the number of cases for each class, and assigns the new case to the same class to which most of its neighbours belong [158].

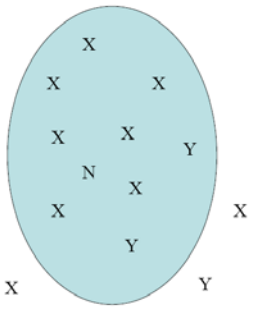

Figure 2.3: N is a new case. It would be assigned to the class X because the seven X's within the ellipse outnumber the two Y's (*cf.* also [158]).

Many researchers have found that the *k-NN* algorithm achieves very good performance in their experiments on different data sets [6, 75, 170].

One of the drawbacks of *k-NN* algorithm is its efficiency, as it needs to compare a test document with all samples in the training set. In addition, the performance of this algorithm greatly depends on two factors, that is, a suitable similarity function, such as the cosine function, and an appropriate value for the parameter *k* [5]. In the traditional *k-NN* algorithm, the value of *k* is fixed beforehand. If *k* is too large, big

classes will overwhelm small ones. On the other hand, if *k* is too small, the advantage of *k-NN* algorithm, which could make use of many experts, will not be exhibited. In practice, the value of *k* is usually optimized by many trials on the training and validation sets. One interesting problem is how to pick the best value for *k*. Some researchers use $k = 20$, whereas others found $30 \leq k \leq 45$ to yield the best effectiveness. Various experiments have shown that increasing the value of *k* does not significantly degrade the performance [45].

k-NN is a traditional statistical pattern recognition algorithm. It has been studied extensively for text categorization applications. In essence, *k-NN* makes a prediction based on the *k* training patterns closest to unseen (test) pattern, according to distance metric [66, 90]. The most common distance function is *Euclidean distance*, which represents the usual manner in which humans think of distance in the real world:

$$d_{Euclidean}(x, y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \tag{2.3}$$

where $x = (x_1, x_2, ..., x_m)$ and $y = (y_1, y_2, ..., y_m)$ represent the m attribute values of two records [90].

To determine the class of a new example *E*:

- Calculate the distance between *E* and all examples in the training set.
- Select *k*-nearest examples to *E* in the training set.
- Assign *E* to the most common class among its *k-NN*.

Example

In the figure below (Figure 2.4) we have three classes and the goal is to find a class label for the unknown example x_u. In this case we use the Euclidean distance and a value of $k = 5$ neighbours. Of the 5 closest neighbours, 4 belong to ω_1 and 1 belongs to ω_3, so, x_u is assigned to ω_1, the predominant [57].

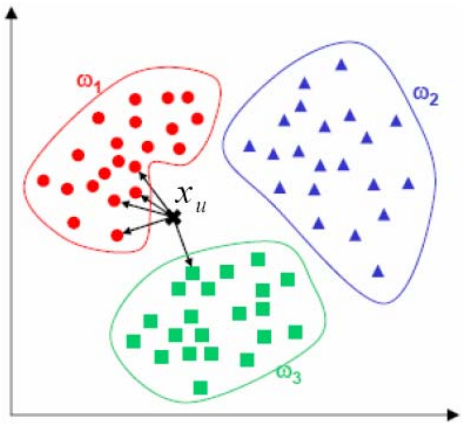

Figure 2.4: k-NN example (*cf.* also [57]).

For text data, the cosine similarity metric is often effective [15].

The algorithm assumes that the instances (the documents) are represented as points in a Euclidian multidimensional space. To classify a new instance, one must find first the instances from the training data that are nearest to it. To find these instances, a measure of similarity is used. We then compute a score for each category of the near documents. At the end, the scores are filtered by a threshold value and the document is labeled with the remaining categories. The similarity measure between two documents is computed as the cosine value between the corresponding normalized document vectors belonging to *t*-dimensional space:

$$simil(d_i, d_j) = \cos(d_i, d_j) = \frac{\sum_{k=1}^{t} w_{d_i k} \times w_{d_j k}}{\sqrt{\sum_{k=1}^{t} w_{d_i k}^2 \times w_{d_j k}^2}}, \qquad (2.4)$$

where $w_{d_i k}$ is the weight of word k of document d_i, $w_{d_j k}$ is the weight of word k of document d_j, and for the weight, (*Tfidf*) scheme can be chosen (see Section 2.3.1). For the evaluation of the score, the algorithm of similarity summing can be chosen. For one category, the score is defined as the sum of the similarities between the new document (d) and the k-nearest neighbour documents that contain the current category among their categories. The formula for this definition is the following:

$$Score(c_i, d) = \sum_{d' \in D_{c_i}'} simil(d, d'), \qquad (2.5)$$

where $D_{c_i}', i = 1,2,...,n;$ represents the set of nearest neighbours that contain the category c_i. Many implementations use this formula for computing the score [82, 132].

The Algorithm

1. Input: new document d;
2. Training collection: $D = \{d_1, d_2, ..., d_n\}$;
3. Predefined categories: $C = \{c_1, c_2, ..., c_n\}$;
4. Compute similarities:

   for $(d_i \in D)$, $simil(d, d_i) = \cos(d, d_i), i = 1,2,...,n;$ where the cosine value between the corresponding normalized document vectors is computed as:

$$simil(d, d_i) = \cos(d, d_i) = \frac{d \cdot d_i}{\|d\|_2 \times \|d_i\|_2} = \frac{\sum_{k=1}^{n} w_{dk} \times w_{d_i k}}{\sqrt{\sum_{k=1}^{n} w_{dk}^2 \times w_{d_i k}^2}}.$$

5. Select *k*-nearest neighbour:

   Construct *k*-document subset D_k so that

   $simil(d, d_i) < \min(simil(d, doc) \mid doc \in D_k) \ \forall d_i \in D - D_k$.

6. Compute score for each category:

$$Score(c_i, d) = \sum_{d' \in D_{c_i}'} simil(d, d')$$

   where $D_{c_i}', i = 1,2,...,n;$ represents the set of nearest neighbours that contain the category c_i.

7. Output: Assign to *d* the category c_i with the highest score.

[82, 85, 132]

E. Han and G. Karypis [59], proposed the Centroid-based* classifier and showed that it gives better results than other known text categorization methods. Other group of researchers has shown that removing outliers (the elements whose similarity to the

centroid of the corresponding category is below a threshold) from the training categories significantly improves the classification results obtained with *k-NN* method. Their experiments show that the new method gives better results than the Centroid-based classifier. The group observed that the training data items that are far away from the center of its training category reduce the accuracy of classification. Their hypothesis is that those items represent noise and not useful training examples and thus decrease the classification accuracy, and consequently they are excluded from consideration (see Figure 2.5). Specifically, at the training stage the center C_i of each category S_i using (2.4) is calculated. Then, new categories are formed by discarding outliers:

$$S'_i = \{d \in S_i : simil(d, C_i) > \varepsilon\}, \tag{2.6}$$

where the choice of the threshold ε was done according to the best accuracy value obtained by testing with respect to different ε values, and an automatic choice of the threshold value ε is to be considered. Finally, the *k-NN* classifier, using these modified categories, is applied.

The improved *k-NN* text categorization method has shown almost 10% better accuracy than the original Centroid-based classifier, which was reported in [59] as the most accurate text categorization method [59, 140].

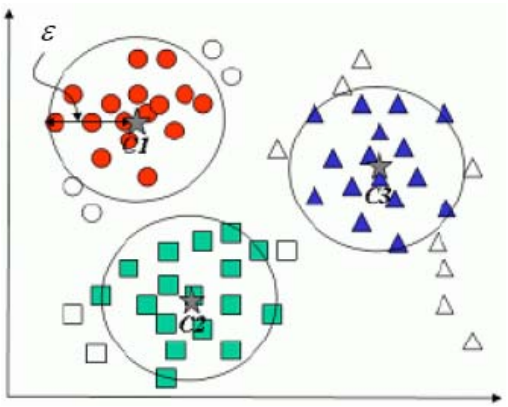

Figure 2.5: Example of classification [140].

***Centroid-based classifier:** Given a set S_i of documents, the *i-th* training category, its center is defined as its average vector:

$$C_i = \frac{1}{|S_i|} \sum_{d \in S_i} d, \tag{2.7}$$

where $|S_i|$ is the number of documents in the category. For a new data item the category is chosen that maximizes the similarity between the new item and the centers of each category. This was reported as the best known classifier so far [59, 140].

The Algorithm

1. Input: new document $d = (w_1, w_2, ..., w_n)$; where w_i is a term weight, $\forall i$.
2. Predefined categories: $C = \{c_1, c_2, ..., c_n\}$;
3. Compute centroid vector:

$$c_i = \frac{\sum_{d' \in c_i} d'}{|c_i|}, \forall c_i \in C,$$

   where $|c_i|$ is the number of documents in the category.

4. Similarity model - cosine function:

$$simil(d_i, d_j) = \cos(d_i, d_j) = \frac{d_i \cdot d_j}{\|d_i\|_2 \times \|d_j\|_2} = \frac{\sum_{k=1}^{n} w_{d_i k} \times w_{d_j k}}{\sqrt{\sum_{k=1}^{n} w_{d_i k}^2 \times w_{d_j k}^2}}.$$

5. Compute similarity:

$$simil(c_i, d) = \cos(c_i, d).$$

6. Output:

   Assign to document *d* the category $c_{\max}$ that maximizes the similarity between the new item and the centers of each category:

$$simil(c_i, d) \leq simil(c_{\max}, d).$$

[85, 140]

2.1.3 Support Vector Machines

Support Vector Machines (*SVMs*) were introduced by V. Vapnik in 1995 [7, 161] and utilized for text categorization by Joachims in 1998 [75, 76, 134]. *SVMs* have proven to be fast effective classifiers for text documents and solve the problem of dimensionality [38]. In fact, it is known that *SVMs* are capable of effectively processing feature vectors of some 10000 dimensions, given that these are sparse [94]. It is interesting to note that the very strong theoretical background of *SVMs* did not make them widely appreciated at the beginning. The publication of the first papers by Vapnik, Chervonenkis and co-workers in 1964/65 went largely unnoticed till 1992. This was due to a widespread belief in the statistical and/or machine learning community that, despite being theoretically appealing, *SVMs* are neither suitable nor relevant for practical applications. They were taken seriously only when excellent results on practical learning benchmarks were achieved in digit recognition, computer vision and text categorization [150].

2.1.3.1 Binary Classifier (Separable Case)

Given a set of separable data points x_i $(i = 1,2,...,m)$, where $x_i \in \mathbf{R}^n$, with labels $y_i \in \{-1,1\}$ and a hyperplane,

$$(w \cdot x) + b = 0, \tag{2.8}$$

(where $w \in \mathbf{R}^n$ is the normal to the plane, $b \in R$ is the distance from the origin and $x \in \mathbf{R}^n$) corresponding to decision functions:

$$f(x) = \text{sign}((w \cdot x) + b). \tag{2.9}$$

The inputs x_i assigned to the positive class, if $f(x) \geq 0$, and otherwise to the negative class. The set of vectors is said to be optimally separated by the hyperplane if it is separated without error and the distance between the closest vectors to the hyperplane is maximal. The *margin* (Figure 2.6), measured perpendicularly to the hyperplane, equals $2/\|w\|_2$. This can be seen by considering two points x_1, x_2 on opposite sides of the margin, i.e.:

$$(w \cdot x_1) + b = 1, \tag{2.10}$$

$$(w \cdot x_2) + b = -1, \tag{2.11}$$

and projecting them onto the hyperplane normal vector $w/\|w\|_2$. Then the problem turned into constrained optimization problem:

$$\text{maximize } 2/\|w\|_2 \text{ or minimize } \|w\|_2^2 \tag{2.12}$$

subject to:

$$y_i \cdot ((w \cdot x_i) + b) - 1 \geq 0 \ (i = 1,2,...,m). \tag{2.13}$$

[19, 32, 63, 74, 77, 104, 109, 129, 131, 148]

A way to solve (2.12) is through its Lagrangian dual [19]:

$$\max_{\alpha \geq 0} \ (\min_{w,b} L(w,b,\alpha)), \tag{2.14}$$

where

$$L(w,b,\alpha) = \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{m} \alpha_i (y_i \cdot ((x_i \cdot w) + b) - 1), \tag{2.15}$$

and α_i are the Lagrange multipliers. The Lagrangian L has to be minimized with respect to the *primal variables* w and b and maximized with respect to the *dual variables* $\alpha_i \geq 0$ [56, 104]. The corresponding dual is found by differentiating with respect to w and b:

$$\frac{\partial L(w,b,\alpha)}{\partial w} = w - \sum_{i=1}^{m} y_i \alpha_i x_i = 0, \tag{2.16}$$

$$\frac{\partial L(w,b,\alpha)}{\partial b} = \sum_{i=1}^{m} y_i \alpha_i = 0, \tag{2.17}$$

and re-substituting the relations obtained:

$$w = \sum_{i=1}^{m} y_i \alpha_i x_i, \tag{2.18}$$

$$0 = \sum_{i=1}^{m} y_i \alpha_i, \tag{2.19}$$

into the primal to obtain:

$$\begin{aligned} L(w,b,\alpha) &= \frac{1}{2}(w \cdot w) - \sum_{i=1}^{m} \alpha_i (y_i \cdot ((w \cdot x_i) + b) - 1) \\ &= \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j) - \sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j) \end{aligned}$$

$$
\begin{aligned}
&+\sum_{i=1}^{m}\alpha_i \\
&=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\ \sum_{j=1}^{m}y_i y_j\alpha_i\alpha_j(x_i\cdot x_j) \\
&=W(\alpha), \qquad (2.20)
\end{aligned}
$$

which is the (Wolfe) dual variables Lagrangian. In order to find the optimal hyperplane, a dual Lagrangian has to be maximized with respect to non-negative α_i, and with respect to equality constraints as follows:

$$\alpha_i \geq 0 \ (i=1,2,...,m), \qquad (2.21)$$

$$\sum_{i=1}^{m} y_i\alpha_i = 0. \qquad (2.22)$$

Using the solution of this problem the decision function can be written as:

$$f(x)=\text{sign}\,(\sum_{i=1}^{m}\alpha_i y_i(x_i\cdot x)+b). \qquad (2.23)$$

According to the Karush-Kuhn-Thucker (*KKT*) conditions:

$$\alpha_i(y_i\cdot((w\cdot x_i)+b)-1)=0 \ (i=1,2,...,m). \qquad (2.24)$$

the non-zero α_i correspond to:

$$y_i\cdot((w\cdot x_i)+b)=1. \qquad (2.25)$$

[25, 52, 77, 81, 120, 136]

It means that the vectors which lie on the margin play the crucial role in the solution of the optimization problem. Such vectors are called support vectors [73]. Points that are not support vectors do not influence the position and orientation of the separating hyperplane and do not contribute to the hypothesis [103].

In the Wolfe dual representation of *SVMs* problem, datapoints appear only in the form of dot product in both the objective function and the solution. By replacing inner products with a kernel function to project the data into a higher dimensional feature space, very flexible representations can be obtained, where the two classes of data are more readily separable [14, 26, 63, 86,131]:

$$x_i\cdot x_j \rightarrow \phi(x_i)\cdot\phi(x_j). \qquad (2.26)$$

A popular choice of kernel is the Gaussian radial basis function:

$$K(x,z)=\exp(-0.5\|x-z\|_2^2/2\sigma^2). \qquad (2.27)$$

So, for a given choice of kernel, the learning task therefore involves maximization of the objective function [3, 14. 19, 32, 103, 120]:

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j). \tag{2.28}$$

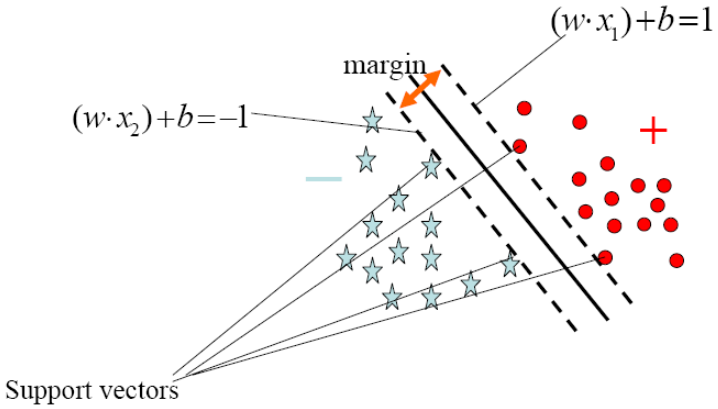

Figure 2.6: Support vector machines.

2.1.3.2 Soft Margin for Non-Separable Case

To deal with cases where there may be no separating hyperplane due to noisy labels of both positive and negative training examples, the soft margin *SVMs* is proposed, which is formulated as:

$$\text{minimize:} \frac{1}{2}(w \cdot w) + C \sum_{i=1}^{m} \xi_i \tag{2.29}$$

subject to:

$$y_i \cdot ((w \cdot x_i) + b) \geq 1 - \xi_i, \ \xi_i \geq 0 \ (i = 1,2,...,m), \tag{2.30}$$

where $C \geq 0$ is a parameter that controls the amount of training errors allowed and $\xi_i \geq 0, \forall i,$ are slack variables [93, 96].

Hence, for such a *generalized* optimal separating hyperplane, the functional to be minimized comprises an extra term accounting the cost of overlapping errors. In fact, the cost function (2.29) can be even more general as given below:

$$\text{minimize:} \ \frac{1}{2} w \cdot w + C \sum_{i=1}^{m} \xi_i^k, \tag{2.31}$$

subject to same constraints . This is a convex programming problem that is usually solved only for $k = 1$ or $k = 2,$ and such soft margin *SVMs* are dubbed L_1 and L_2

SVMs respectively. For L_1 *SVMs* $(k = 1)$, the solution to a quadratic programming problem is given by the primal Lagrangian shown below:

$$L(w,b,\alpha,\xi) = \frac{1}{2} w \cdot w + C\sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i (y_i \cdot ((w \cdot x_i) + b) - 1 + \xi_i) - \sum_{i=1}^{m} r_i \xi_i \qquad (2.32)$$

with Lagrange multipliers $\alpha_i \geq 0$ and $r_i \geq 0$. The corresponding dual is found by differentiating with respect to w, b and ξ,

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0, \qquad (2.33)$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{m} \alpha_i y_i = 0, \qquad (2.34)$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - r_i = 0, \qquad (2.35)$$

and re-substituting the relations obtained into the primal; we obtain the following adaptation of the dual objective function

$$L(w,b,\xi,\alpha,r) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j), \qquad (2.36)$$

subject to:

$$0 \leq \alpha_i \leq C, \qquad (2.37)$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0. \qquad (2.38)$$

Once the α_i's are obtained the other primal variables w, b, ξ and r can be easily determined using the *KKT* conditions [14, 19, 72, 81, 107].

2.1.3.3 Multi-Class Classifier

So far, we have talked about binary classification, where the class labels can only take two values: ± 1. Many real-world problems, however, have more than two classes, i.e., involve multi-class classification and a lot of different approaches have been proposed to do this. Yielding better results than others, most approaches for multi-class *SVMs* decompose the data set to several binary problems. The "one versus one" and "one versus all" are two well-known approaches dealing with this issue [3, 19, 104, 135].

One Versus All

In the "one versus all" method it is common to construct a set of binary classifiers $f^1,..., f^k$, to get *k-class classifiers*, each trained to separate one class from the rest by labelling all the examples in the *i-th* class as positive, and the rest examples as negative [103, 104, 152].

One Versus One

In "one versus one" approach we train a classifier for each possible pair of classes. For a *k*-class problem, this results in $k(k-1)/2$ binary classifiers. In the prediction stage, a voting strategy is used where the testing point is designated to be in a class with the maximum number of votes. This approach has been found to be effective compared to the "one versus all" approach [13, 19, 104, 135].

2.2 Text Preprocessing

The classification of textual data slightly differs from that of numerical data, in view of the preprocessing that is carried out in the case of the former [157]. First of all, documents need to be preprocessed. This usually means stopword filtering for omitting meaningless words (e.g., a, an, this, that), word stemming for reducing the number of distinct words, lowercase conversion etc.. Then the transformation takes place [123].

2.2.1 Filtering, Lemmatization and Stemming

In order to reduce the dimensionality of the description of documents within a collection, the set of words describing the documents can be reduced by filtering and lemmatization or stemming methods.

Filtering methods remove words from the documents. A standard filtering method is stopword filtering. The idea of stopword filtering is to remove words that bear little or no information content, like articles, conjunctions, prepositions, etc.. Furthermore,

words that occur extremely often can be said to be of little information content to distinguish between documents, and also words that occur very seldom are likely to be of no particular statistical relevance and can be removed [69,92].

Lemmatization methods try to map verb forms to the infinitive tense and nouns to the singular forms. However, in order to achieve this, the word form has to be known, i.e., the part of speech of every word in the text document has to be assigned. Since this tagging process is usually quite time consuming and still error-prone, in practice frequently stemming methods are applied.

Stemming methods try to build the basic forms of words, i.e., strip the plural 's' from nouns, the 'ing' from verbs, or other affixes. A stem is a natural group of words with equal (or very similar) meaning. After the stemming process, every word is represented by its stem [69]. One of the common stemming algorithms is the Porter's algorithm [69,92]. He defined a set of production rules to iteratively transform (English) words into their stems [69].

In the experimental part, the preprocessing carried out on the textual data implies the removal of stopwords and application of Porter's stemming algorithm.

2.3 Document Representation

In order for the textual records to be used for classification, they need to be converted to a numeric format and represented in matrix form. This conversion process is known as *document representation* [103]. The most commonly used document representation is the so called *vector space model*. In this model [157], each document is represented as a vector. Each dimension in the vector stands for a distinct 'unit of content' of a document in the 'content unit' space of the document collection [118], where each unit of content could be a single word, phrase, part of a word or even a 'concept'.

Consider a document collection, where the units of content are single terms. Assume there are only three distinct terms, α, β and γ, in a document. Then we have a three dimensional vector. Say in the first document, $D1$, term 1 occurs only once and term 3 occurs only twice. (1,0,2) is one possible representation of this document which accounts for the term frequency of the words. (1,0,1) is another representation, which

is a vector showing the existence of a word. Such variations are generally referred to as various *weighting schemes* [103].

2.3.1 Term Weighting Schemes

Each of the weighting schemes captures different features of a document; some more, some less. More would be mentioned in the subsections below.

There are several ways of determining the weight a_{ik} of the word i in document k, but most of the approaches are based on two empirical observations regarding the text:

- The more times a word occurs in a document, the more relevant it is to the topic of the document.
- The more times the word occurs throughout all documents in the collection, the more poorly it discriminates between documents [103, 118, 157]. Let:

Term frequency, f_{ij} be the frequency of term i in document j

Document frequency, df_i the number of documents in which term i occurs

N – number of documents and

M – number of terms

2.3.1.1 Binary Weighting

This is the simplest approach, also called boolean weighting, in which the weight is considered to be 1 if the word occurs in the document and 0 otherwise [103, 118, 157]:

$$a_{ik} = \begin{cases} 1, \text{ if } f_{ik} > 0 \\ 0, \text{ otherwise} \end{cases} \tag{2.39}$$

2.3.1.2 Term Frequency Normalized

This approach uses the frequency of the word in the document. It normalizes the length of each term to 1. This has the effect of giving high weight to infrequent

terms. Such a weighting only depends on the sum of square frequencies and not the distribution of those frequencies [103, 157]:

$$a_{ik} = \frac{f_{jk}}{\sqrt{\sum_{j=1}^{N}\left[f_{ij}\right]^2}} \tag{2.40}$$

2.3.1.3 Term Frequency Inverse Document Frequency

A well-known approach, that takes into account the frequency of the word throughout all documents in the collection is the term frequency inverse document frequency weighting (*Tfidf*), which assigns the weight to word i, in proportion to the number of occurrences of the word in the document, and in inverse proportion to the number of documents in the collection for which the word occurs at least once. It is given as [103, 157]:

$$a_{ik} = f_{ik} \times \log\left(\frac{N}{df_i}\right). \tag{2.41}$$

This approach is adopted in various experimental studies in this thesis to gain better results.

2.3.1.4 Term Frequency Inverse Document Frequency-Length Normalized

Same as the previous weighting, except that the document feature vector is normalized to unit length. This takes into account the fact that documents might be of different lengths [173]:

$$a_{ik} = \frac{f_{ik} \times \log\left(\frac{N}{df_i}\right)}{\sqrt{\sum_{j=1}^{M}\left[f_{jk} \times \log\left(\frac{N}{df_j}\right)\right]^2}}. \tag{2.42}$$

2.4 Feature Selection

A major characteristic, or difficulty, of text categorization problems is the high dimensionality of text data. Even for a moderate-sized text collection, the feature space consists of documents, which can be tens or hundreds of thousands of terms. This is unacceptably high for many learning algorithms [92, 112, 171]. As such, feature selection (*FS*) or *Term Space Reduction* (*TSR*) is widely used for reasons of both efficiency and efficacy, when applying machine learning methods to text categorization [35, 48, 100, 143]. In fact, feature selection is a very important step in the categorization process [78, 114]. It is proved to improve both the speed and the quality of categorization; because irrelevant and redundant words often degrade the performance of categorization algorithms both in speed and categorization accuracy [114, 137]. To reduce the number of features, we first remove features based on overall frequency counts, and then select a small number of features based on their fit to categories [35]. Various feature selection methods have been proposed and successfully applied for the reduction of the feature set without sacrificing text categorization performance. They include *document frequency*, *mutual information*, *information gain*, *OddsRatio*, and *chi-square statistic* [92, 171]. Since the effectiveness of each method in text categorization is affected by the characteristics of the test corpus and the chosen machine learning algorithm; the criteria for the choice of feature selection methods are not clear [92]. A difficulty with most complicated feature selection methods is that they are very time-consuming and so it is not practical or possible to perform the feature selection process whenever new training examples are available [92]. Although feature selection is primarily performed to select relevant and informative features, it can have other motivations, including [58]:

1. general data reduction, to limit storage requirements and increase algorithm speed;
2. feature set reduction, to save resources in the next round of data collection or during utilization;
3. performance improvement, to gain in predictive accuracy;
4. data understanding, to gain knowledge about the process that generated the data or simply visualize the data.

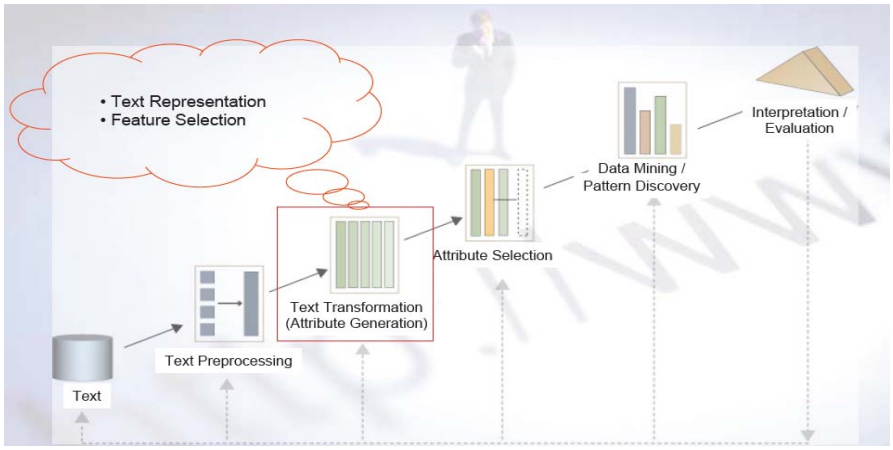

Figure 2.7: Text mining process [20].

2.4.1 Information Gain

In the text domain, the most popular used *FS* algorithms are still the traditional ones such as *Information Gain* (*IG*), χ^2-test (*CHI*) (this approach is implemented in various experimental studies in this thesis), Document Frequency (*DF*) and Mutual Information (*MI*), etc.. Information gain is frequently employed as a term-goodness criterion in the field of machine learning. It measures the number of bits of information obtained for category prediction by knowing the presence or absence of a word in a document. Given a corpus of training text, we compute the information gain of each term, and then remove those features whose information gain was less than some pre-determined threshold [112, 169, 171, 176].

The information gain of a word *w* is defined to be:

$$IG(w) = -\sum_{j=1}^{k} P(c_j)\log P(c_j) + P(w)\sum_{j=1}^{k} P(c_j|w)\log P(c_j|w) + P(\overline{w})\sum_{j=1}^{k} P(c_j|\overline{w})\log P(c_j|\overline{w}) \tag{2.43}$$

where $c_1, c_2, ..., c_k$ denote the set of categories, $\overline{w}$ the absence of word *w*, $P(w)$ is the probability that the word w occurs in any randomly selected document, $P(c_j)$ is the probability that any randomly selected document belongs to class c_j, and $P(c_j|w)$ is the conditional probability of finding a document belonging to class c_j, given that it contains word *w* [42, 112, 171].

The lower the information gain the less important is the feature. So, information gains of common terms are very low and have no chance in selection process [103].

2.4.2 Chi-square Statistic

Chi-square statistic (*CHI*) is the most commonly used method of comparing proportions. It checks whether there is a relationship between being in one of two groups and a characteristic under study; where the groups would be 1) the documents from a category c, 2) all other documents and the characteristic would be: "document contains term t" [111]. In other words, *CHI* measures the lack of independence between a term (word) w and a category c [176]. Using the two-way contingency table of term t and a category c, where *A* is the number of times *t* and c co-occur, *B* is the number of times the t occur without c, C is the number of times *c* occurs without t, D is the number of times neither c nor *t* occurs, and *N* is the total number of documents (i.e., $N = A + B + C + D$), the term-goodness measure, a simple formula for *CHI* is defined to be:

$$\chi^2(t,c) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)}, \tag{2.44}$$

Here we computed for each category the *CHI* between each unique term in a training corpus and that category. For multiple categories we compute *CHI* for each category and then combine the category-specific scores of each term into two scores:

- we can require to discriminate well across all categories, then we need to take the expected value of χ^2 [111, 171]:

$$\chi^2_{avg}(t) = \sum_{j=1}^{k} P(c_j)\chi^2(t,c_j), \tag{2.45}$$

- or to discriminate well for a single category, then we take the maximum:

$$\chi^2_{\max}(t) = \max_{j=1}^{k} \{\chi^2(t,c_j)\}, \tag{2.46}$$

where $c_1, c_2, ..., c_k$ denote the set of categories.

CHI has a natural value of zero if *t* and *c* are independent. It is known that *CHI* is unreliable for low frequency terms and is computationally expensive [171].

Example

Is "jaguar" a good predictor for the "auto" class?

Table 2.1.1: The contingency table of the term 'jaguar' and the category 'auto'[111].

	Term = jaguar	*Term ≠ jaguar*
Class = auto	2	500
Class ≠ auto	3	9500

We want to compare:

- the observed distribution above, and
- null hypothesis: that jaguar and auto are independent.

Under the null hypothesis: (*jaguar* and *auto* – independent):

How many co-occurrences of *jaguar* and *auto* do we expect?

- We would have: $P(j,a) = P(j) \times P(a)$.
- So, there would be: $N \times P(j,a)$, i.e., $N \times P(j) \times P(a)$.
- $P(j) = (2+3)/N; P(a) = (2+500)/N; N = 2+3+500+9500$.
- Thus: $N \times (5/N) \times (502/N) = 2510/\mathrm{N} = 2510/10005 \approx 0.25$.

Table 2.1.2: The contingency table of the term 'jaguar' and the category 'auto'(*cf. also* [111]).

	Term = jaguar	*Term ≠ jaguar*
Class = auto	0.25	502
Class ≠ auto	4.75	9498

Table 2.1.3: The contingency table of the term 'jaguar' and the category 'auto'(cf. also [111]).

	Term = jaguar		*Term ≠ jaguar*	
	observed f_o	*expected* f_e	*observed* f_o	*expected* f_e
Class = auto	2	(0.25)	500	(502)
Class ≠ auto	3	(4.75)	9500	(9498)

χ^2 is interested in $((f_o - f_e)^2 / f_e$ summed over all table entries:

$$\chi^2(j,a) = \sum (O-E)^2 / E = (2-.25)^2/.25 + (3-4.75)^2/4.75 + (500-502)^2/502 + (9500-9498)^2/9498 = 12.9 \quad (p < .001).$$

The null hypothesis is rejected with confidence .999, since 12.9 > 10.83 (the value for .999 confidence) [111].

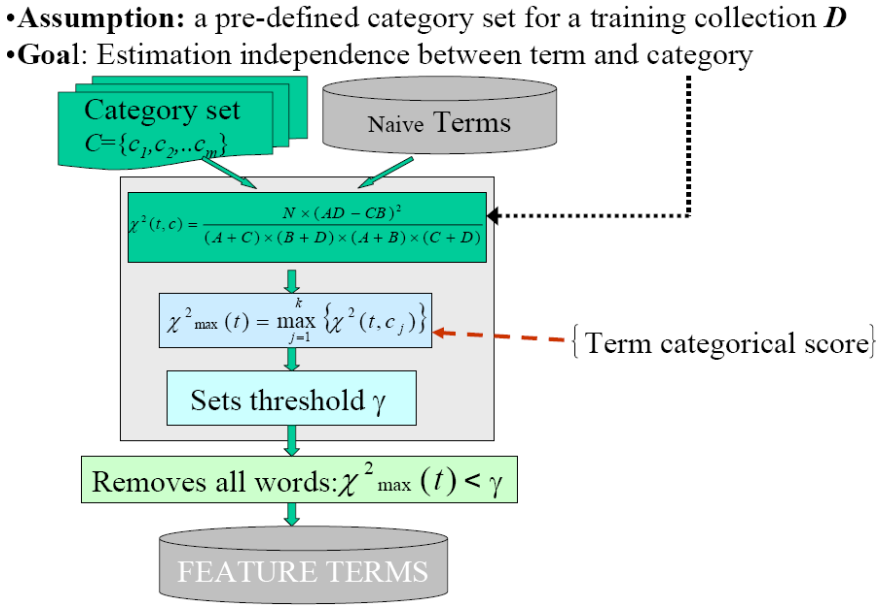

Figure 2.8: The chi-square statistic algorithm (*cf. also* [85]).

2.4.3 Markov Blanket Algorithm

The *Markov Blanket* (*MB*) feature selection algorithm introduced by Koller and Sahami is based on ideas from information theory and probabilistic reasoning [4].

Let F be a full set of features, F_i a feature, and $M \subset F$ $(F_i \notin M)$, M is said to be *a Markov Blanket* for F_i iff:

$$P(F - M - \{F_i\}, C \mid F_i, M) = P(F - M - \{F_i\}, C \mid M), \qquad (2.47)$$

where C is a class variable. The Markov Blanket condition requires that M subsumes not only the information that F_i has about C but also about all of the other features [173]. Let f be an assignment of values $(f_1, f_2, ..., f_n)$ to F, G be some subset of F and f_G be the value vector of G. In general, the goal of feature selection can be formalized as selecting a minimum subset G such that $P(C \mid G = f_G)$ is equal or as close as possible to $P(C \mid F = f)$, where $P(C \mid G = f_G)$ is the probability distribution of different classes given the feature values in G and $P(C \mid F = f)$ is the original distribution given the feature values in F [173]. As distance metric, the information-theoretic measure of *cross-entropy* (also known as KL-distance) is used. The cross-entropy of two distributions μ and σ over some probability space Ω, is defined as:

$$D(\mu, \sigma) = \sum_{x \in \Omega} \mu(x) \log \frac{\mu(x)}{\sigma(x)}. \qquad (2.48)$$

Here μ is the 'right' distribution, and σ is the approximation to it. Then $D(\mu, \sigma)$ measures the extent of the 'error' that we make by using σ as a substitute for μ. The above equation can be transformed to be used for feature selection as follows:

$$\mu = P(C \mid F = f), \qquad (2.49)$$

$$\sigma = P(C \mid G = f_G), \qquad (2.50)$$

and Ω is the set of possible classification $\{c_1, c_2, ..., c_n\}$. Therefore, we define:

$$\delta_G(f) = D(P(C \mid F = f), P(C \mid G = f_G)). \qquad (2.51)$$

In order to have a metric which allows us to compare one feature set G to another, we must integrate values of $\delta_G(f)$ for different feature vectors f into a single quantity. This is done as follows:

$$\text{Expected cross-entropy, } \Delta_G = \sum_f P(f)\delta_G(f) \tag{2.52}$$

In wanting to choose a reduced dataset G to approximate F, we must minimize Δ_G. Assume that G is the current set of features ($G = F$ in the beginning). At any phase, if there exists a Markov blanket M for F_i within the current G, F_i is removed from G [89, 173]. Unfortunately, there might not be a full Markov blanket for a feature, but rather only an approximate one that subsumes the information content of the feature. The approximation can be done by observing that, if M_i is really a Markov Blanket for F_i, then:

$$D(P(C \mid M_i = f_{M_i}, F_i = f_i), P(C \mid M_i = f_{M_i})) = 0, \tag{2.53}$$

for any assignment of feature values f_{M_i} and f_i to M_i and F_i respectively, where, M_i is the candidate Markov Blanket for F_i. We therefore define the expected cross-entropy:

$$\delta_G(F_i \mid M_i) = \sum_{f_{M_i}, f_i} P(M_i = f_{M_i}, F_i = f_i) \cdot D(P(C \mid M_i = f_{M_i}, F_i = f_i), P(C \mid M_i = f_{M_i})). \tag{2.54}$$

If M_i is, in fact, a Markov Blanket for F_i, then, $\delta_G(F_i \mid M_i) = 0$, and if it is an approximate Markov Blanket, then this value will be low. The feature F_i which has the minimal $\delta_G(F_i \mid M_i)$, will be eliminated. These approximations result in the following algorithm [22, 83]:

1. Compute the cross-entropy of the class distribution given pairs of features, $\gamma_{ij} = D(P(C \mid F_i = f_i, F_j = f_j), P(C \mid F_j = f_j))$ of every pair of features F_i and F_j.
2. Define $G = F$.
3. For each feature $F_i \in G$, let M_i be the set of K features F_j in $G - \{F_i\}$ for which γ_{ij} is smallest.
4. Compute $\delta_G(F_i \mid M_i)$, for each i.
5. Choose the i for which this quantity is minimal, and define $G = G - \{F_i\}$.
6. Iterate steps 3 onwards until a certain pre-specified number of features have been eliminated.

Theoretically, the *MB* algorithm requires $O(n^2(m + \log n))$ operations for computing the correlation matrix and sorting it, whilst the subsequent feature selection process requires $O(r \cdot n \cdot k.m \cdot 2^k \cdot c)$ time, where n is the initial number of features, m the number of instances, r the number of features to be eliminated, k, the small, fixed number of conditioning features, and c is the number of classes. The Markov Blanket criterion aims to only remove attributes that are unnecessary. Such attributes are either totally irrelevant to the target concept, or are redundant given other attributes. It is one of the few feature selection techniques that is able to deal with both these types of unnecessary features [83, 103].

2.5 Feature Extraction

There are two major types of *dimension reduction techniques* (*DRT*), feature transformation (also known as Feature Extraction (*FE*)), (the features in reduced feature set are not of the same type of the features in original feature set but are obtained by combinations or transformations of the original ones), and feature selection (reduced feature set is a subset of the original feature set) [27, 115, 154, 169]. The traditional *FE* algorithms reduce the dimension of data by linear algebra transformations (such as *Principal Component Analysis* (*PCA*), *Linear Discriminant Analysis* (*LDA*) and *Maximum Margin Criterion* (*MMC*), etc.) or nonlinear transformations (*Locally Linear Embedding* (*LLE*), *ISOMAP*, etc.). Though the *FE* algorithms have been proved to be very effective for dimension reduction, the high dimension of data sets in the text domain often fails many *FE* algorithms due to their high computational cost [169]. Widely used examples include, Principal Components Analysis (*PCA*), *Factor Analysis*, *Projection Pursuit*, Latent Semantic Indexing (*LSI*), *Independent Component Analysis* (*ICA*), and *Random Projection* (*RP*). In the following, the LSI approach is described [47, 154].

2.5.1 Latent Semantic Indexing

Latent Semantic Indexing (*LSI*) is an efficient dimension reduction technique that has been developed to analyze text based on the popular *Singular Value Decomposition*

(*SVD*), and the text analysis using this method is called *Latent Semantic Analysis* (*LSA*). As the name implies, this technique helps in extracting the hidden semantic structure from the text rather than just the usage of the term occurrences [17, 64, 84, 106]. The idea is that words are semantically similar to the extent that they share contexts. If two words *W* and *Y*, say "tea" and "coffee", frequently occur in the same context *C*, say after "drink", the hypothesis states that *W* and *Y* are semantically related, or, that they are semantically similar. The semantic similarity (or relatedness) of "tea" and "coffee" is thus due to the similarity of usage of these words. This means that the categorization of both words as, for example, referring to hot beverages is only possible because we use them in such a way; that is, for example after the word "drink" and in the neighbourhood of the word "drunk". The categorization is not a cause of usage, but a consequence [130]. By detecting the high-order semantic structure (term-document relationship), it aims to address the ambiguity problem of natural language, i.e., the use of synonymous, and polysemous words, therefore, a potentially excellent tool for automatic indexing and retrieval [154]. The major idea behind *LSI* is to improve the efficacy of a term by document matrix by eliminating semantic noise. This is done, via the *SVD*, by choosing a dimension lower than the effective rank of the matrix and then mapping each term and document as vectors into this lower dimensional space [163].

Let *A* be an *m* terms by *n* documents matrix [17] whose component a_{ij} $(1 \le i \le m)$ is the weight of term t_i in document d_j $(1 \le j \le n)$. The *SVD* of the matrix *A* is given by [22, 151]:

$$A_{m\times n} = U_{m\times r} \sum_{r\times r} V^T_{r\times n} \tag{2.55}$$

Where r is the rank of A $(r \le \min(m,n))$; U and V are column-orthonormal $(U^TU = V^TV = I,$ the identity matrix), and $\sum$ is the diagonal matrix containing singular values of A $(\sum = diag(\sigma_1,\sigma_2,...,\sigma_r))$ which are the nonnegative square roots of eigenvalues of AA^T [17, 22, 154]. Reducing the dimensionality is now done by keeping *k* values in $\sum$ (which are all positive and ordered in decreasing magnitude, i.e., $\sum = diag(\sigma_1,\sigma_2,...,\sigma_k,...,\sigma_r)$, where $\sigma_i > 0$ for $1 \le i \le k$), setting the rest to zero (deleting them) , i.e., $\sigma_i = 0$ for $i > k$. How many values that are to be used

(i.e., the actual number of the constant *k*) equals the number of dimensions one has agreed to settle for. Then, multiplying the retained values with their corresponding columns in *U* and *V* yields the matrix A_k with rank *k*, that is, *A* is approximated as:

$$A_k = U_k \sum_k V_k^T \tag{2.56}$$

in which *A* is projected from *m* dimensional space to *k* dimensional space where $k << m$ (typical values of *m* could be 10,000 to 50,000 or more while *k* can be in the order of 100 or less, literally without significant loss of information), and $\sum_k$ is obtained by deleting the zero rows and columns of $\sum$, and U_k and V_k are obtained by deleting the corresponding rows and columns of *U* and *V* (shown in Figure 2.9) [22, 106, 130, 154]. In the new *k*-dimension, each row of matrix A_k approximately represents one document [154]. The resulting A_k captures the most important associations between terms and documents, and effectively removes noise and redundancy and word ambiguity within the dataset [22, 154].

Solving (2.56) for V_k yields $V_k = A_k^T U_k \sum_k^{-1}$ using the facts that U_k is orthogonal and that $\sum_k$ is symmetric. The *i-th* row of V_k is given by the product of the *i-th* row of A_k^T with the matrix $U_k \sum_k^{-1}$. Using $\vec{q}_o$ ($\vec{q}_o = (q_1, q_2, ..., q_m)$), and q_i is 1 if term *i* is in the query and 0 otherwise) as the *i-th* row of A_k^T yields $\vec{q} = \vec{q}_o U_k \sum_k^{-1}$, which is the representation of $\vec{q}_o$ under the approximation A_k of *A*, and by means of which the *m*-vector $\vec{q}_o$ is mapped into the *k*-vector $\vec{q}$ [22, 85, 130, 154, 163]. In this manner, we may map queries, i.e., the set of keywords to be searched for, sometimes called *pseudo-documents*, into *k*-space and compute the angle between the query and each document in the set. The angle θ between two vectors $\vec{x}$ and $\vec{y}$ is the arccosine of the quotient of their inner product over the product of their respective norms:

$$\cos\theta = \frac{\vec{x}^T \vec{y}}{\|\vec{x}\|_2 \|\vec{y}\|_2}. \tag{2.57}$$

The set of documents to be returned as relevant may be chosen as either those documents with angles to the query vector less than a threshold $\hat{\theta}$, or the *p* documents with the smallest angles [163].

Having specified how the term by document matrix A is approximated in k-space using the $A_k = U_k \Sigma_k V_k^T$ factorization and how user queries are mapped into that space, it is now possible to state the complete algorithm [163].

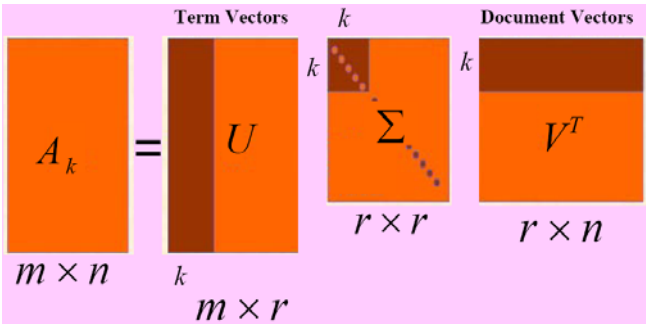

Figure 2.9: Graphical interpretation of the matrix A_k (*cf. also* [88])

The Algorithm

- Let A be the term by document matrix of dimension $m \times n$.
- Factorize $A_{m \times n} = U_{m \times r} \sum_{r \times r} V_{r \times n}^T$ using the *SVD*, observe the rank r of A from the diagonal matrix Σ.
- Choose a dimension $k < r$ and approximate A using the factorization $A_k = U_k \Sigma_k V_k^T$ where U_k is an $m \times k$ orthogonal matrix containing the first k columns of U, Σ_k is a $k \times k$ diagonal matrix containing the first k columns in Σ, and V_k is an $n \times k$ orthogonal matrix containing the first k columns of V.
- For each user query $\vec{q}_o$:
  - Map the query into a k-space vector $\vec{q}$ using $\vec{q} = \vec{q}_o U_k \Sigma_k^{-1}$.
  - Let $D = (\vec{d}_1, \vec{d}_2, ..., \vec{d}_n)^T$. Find $\theta_i = \cos^{-1} \frac{\vec{q} \cdot \vec{d}_i^T}{\|\vec{q}\|_2 \|\vec{d}_i\|_2}$, $\forall i \in \{1,2,...,n\}$.
  - Return either the set $D' \subset D$ that consists of the documents with the angles $\theta_i \leq \hat{\theta}$ for some threshold angle $\hat{\theta}$, OR

- Return the set $D' \subset D$ that consists of the p documents with the smallest angles θ_i.

[163]

2.6 Performance Measures

There are many commonly used performance measures for the evaluation of text classifiers. We need evaluation methods to compare various text classifiers. However, there is no agreement on one single measure for use in all applications. Indeed, the type of measure that is preferable depends on the characteristics of the test data set and on the user's interests [92, 103].

A classifier can be evaluated in terms of precision, recall, accuracy, and error [22, 49]. Recall and precision frequently adopted for the evaluation for the text categorization, are also the standard performance measures for classic information retrieval research [49, 92]. The precision and recall can be estimated in terms of the contingency table for category c_i on a given test set.

Table 2.2: The contingency table for category c_i [22].

a: the number of testing examples correctly assigned to this category. b: the number of testing examples incorrectly assigned to this category. c: the number of testing examples incorrectly rejected from this category. d: the number of testing examples correctly rejected from this category.

From the quantities in table 2.3, precision and recall for a category are defined as [22]:

$$precision = \frac{a}{a+b}, \tag{2.58}$$

$$recall = \frac{a}{a+c}. \tag{2.59}$$

In other words, *precision* is the probability that documents identified as belonging to a class are correctly classified and *recall* is the probability of documents in a class to be correctly identified [49, 78, 92, 118, 127]. Precision has similar meaning as

classification accuracy. But they are difference in that precision considers only examples assigned to the category, while accuracy considers both assigned and rejected cases. Accuracy and error for a category are defined as [22]:

$$accuracy = \frac{a+d}{a+b+c+d}, \tag{2.60}$$

$$error = \frac{b+c}{a+b+c+d}. \tag{2.61}$$

Precision or recall may be misleading when considered alone since they are interdependent. Thus, a combined measure is considered. Among the various combined measures, F_β-measure is one of the most frequently used in text categorization [22, 92]. It is defined as follows [95, 134]:

$$F_\beta = \frac{(1+\beta^2)\times(precision)\times(recall)}{\beta^2\times(precision)+recall}, \tag{2.62}$$

where $0 \le \beta \le \infty$. In equation (2.62), usually $\beta = 1$ is used, which means it gives equal weight to precision and recall and is usually referred to as F_1-measure [22, 49, 95, 134]:

$$F_1 = \frac{(2)\times(precision)\times(recall)}{precision+recall}. \tag{2.63}$$

The above definitions are applicable for each category. To obtain measures relating to all categories, two methods may be adopted: micro-averaging and macro-averaging.

Micro- and Macro-Averaging

Micro-averaging: the performance measures are obtained by globally summing over all individual decisions, i.e.:

$$precision = \frac{\sum_{i=1}^{K} a_i}{\sum_{i=1}^{K}(a_i+b_i)}, \tag{2.64}$$

where K is the number of categories, a_i is the number of testing examples correctly assigned to category i, and b_i is the number of testing examples incorrectly assigned to category i.

Macro-averaging: the performance measures are first evaluated locally for each category and then globally by averaging over the result of the different categories. For precision, this implies:

$$precision = \frac{\sum_{i=1}^{K} precision_i}{K}. \tag{2.65}$$

Recall, accuracy, and error for all categories can be computed similarly, [22, 49].

CHAPTER 3

THE APPLICATION OF TEXT MINING TO QUALITY IMPROVEMENT

There is already more information overflow into most organizations than any one person, or even a large team of people, can mange in a day. Organizations can not afford to ignore this flood of information [46]. It is estimated that unstructured information, such as emails, memos, customer correspondence, and reports, represents 80% of the total business information available to a company [39, 146]. Companies must be able to understand the contents of email messages, contracts, documents, market intelligence, transactions, customer-facing records, etc., that accumulate at an unbearable rate. Without a view into this information, organizations miss finding customer issues, faulty products, fraud, or non-complaint business practices until the problems have become too large to contain. Failing to detect such problems in products or processes puts the entire organization, and quite possibly its customers, at risk [46].

As the amount of unstructured data in our world continues to increase, text mining tools that allow us to sift through this information with ease will become more and more valuable [43]. Given its broad applicability, text mining has seen widespread application in many industries ranging from finance, bioinformatics, pharmaceuticals, telecommunications and others [34, 103].

In this chapter, the capabilities of text mining as a burgeoning quality improvement tool are examined with stating applications in the manufacturing industry and the service sector, followed by experimental studies that illustrate approaches to the applications and manifest making use of various mathematical methods explained in this thesis. The rationale behind the choice of “quality improvement” and these two areas is reasonable: Quality is the most important determinant of profit, and quality improvement increases consumer demand for products and services, [108, [155]. On the other hand, quality improvement is one of the typical ways a firm could benefit

from text mining [23]. Companies that want to improve the quality of their products and/or services need to concentrate their efforts on the proper exploration of data available in their records. By examining data at hand they can discover previously unnoticed patterns of employee behaviour that slow down their processes and diminish the value of their products. By getting to know their customers better, they are more flexible in responding to their needs and reacting to competitors' attempts to undermine their market position [117]. Text mining lets a firm collect data from unstructured sources such as warranty notes, customer contact centers and online information and merge it with data it already has, creating a more powerful knowledge base for better decision making [23]. Services account for more than 75% of the *gross domestic product* (*GDP*) in most developed countries [117]. The share of employment in services continued to rise in virtually all *Organisation for Economic Co-operation and Development* (*OECD*) countries over the 1990s, approaching nearly three-quarters of all jobs in several countries [116]. The U.S. service sector is the largest sector in the economy and accounts for an increasingly significant share of *GDP*. If we define the service sector as the non-manufacturing, nonagricultural, non-mining, and non-construction sectors, it accounted for 78.9 percent of *GDP* in 2002 and for 83 percent of nonagricultural employees [51]. Service industries now dominate the UK economy, although manufacturing remains a key sector. Including government, services account for around 70% of gross domestic product [99]. In 1995, the service sector accounted for two-thirds of *value added* as compared to 27 per cent for production (largely manufacturing industries with 22 per cent) [99].

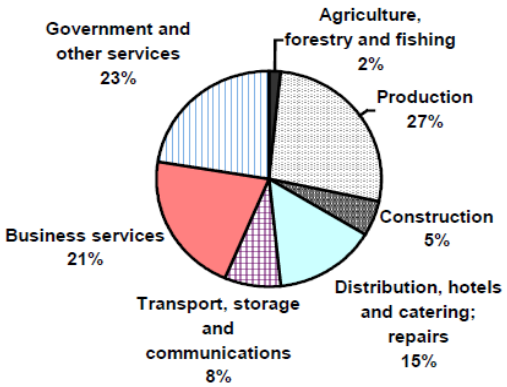

Figure 3.1: Share of output, 1995 [99].

With over 50 per cent contribution to the *GDP*, services form the mainstay of the Indian economy today [141]. The Japanese service sector accounted for about 70% of the nation's *GDP* in 2002 (Figure 3.2) [113].

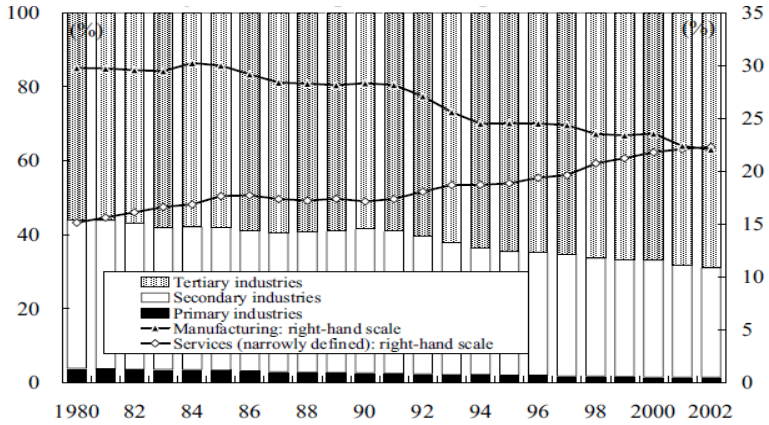

Figure 3.2: Composition of Japan's GDP [113].

The evolution of the service sector has paralleled a decline in the manufacturing sector, which saw its share of *GDP* fall from 29.8% in 1980 to 22.1% in 2002 [113]. In Turkey, the contribution of agriculture, manufacturing and services to Turkish economy are successively 10%, 25%, and 55% [24].

The above-mentioned examples from developed and developing countries have shown that manufacturing and service sector account for more than three forth of *GDP* of these countries. It is therefore important to improve quality in these two vital sectors of the economy: manufacturing and service [117].

3.1 Quality

American Society for Quality defines 'quality' as "a subjective term for which each person has his or her own definition. In technical usage, quality can have two meanings: 1. the characteristics of a product or service that bear on its ability to satisfy stated or implied needs. 2. a product or service free of deficiencies", [2, 164]. The quality of a product or service refers to the perception of the degree to which the

product or service meets the customer's expectations. Various other interpretations for the term quality, such as that given by *ISO* 9000 “Degree to which a set of inherent characteristic fulfills requirements”, can also be introduced [164]. Quality is inherently a part of service, even the word service is in some interpretations synonymous with quality. Thus, in a free market economy with many service providers competing for the customers, the service quality is a critical element that helps creating the extra competitive edge needed for the service providers’ long-term survival [117]. In the manufacturing industry it is commonly stated that “Quality drives productivity”. Improved productivity is a source of greater revenues, employment opportunities and technological advances [141]. Many engineers came to believe that the key to success was a better quality product. As such, improvements in the quality of services and/or production have an impact on the whole economy [29].

3.1.1 A Quality Improvement Tool in Manufacturing

It has been estimated that the amount of information in the world doubles every 20 months. The size and number of databases probably increases even faster [97]. Within this mass of data, it is statistically certain that there is at least one gleaming nugget of knowledge that will increase yields, improve quality, or develop designs [97]. In many manufacturing companies, large databases containing data on failures and repairs of products are maintained, but the knowledge in those databases can be disreputably difficult to identify, capture, and manage [65, 97]. While data mining is designed to handle numerical databases, text mining offers a way of extracting information from textual databases. Although it is estimated that 80% of a company's information is contained in text documents [39, 80, 146, 153], where they can be analyzed to determine and improve the quality of products, there has been a general lack of attention paid to the analysis of textual data compared with that paid to the analysis of numerical ones. This serves as motivation for the work in this thesis [65, 103]. Manufacturing is an application area where text mining can provide significant competitive advantage [65]. Quickly solving product yield and quality problems in a complex manufacturing process is becoming increasingly more difficult [53]. Text

mining technology can increase product yield and quality by quickly finding and solving these problems [53]. The interest in text mining reveals an astute awareness among manufacturing companies regarding the potential of text mining for changing business performance [12]. For example, Ford Motor Company uses text mining technology for early product correction and cost savings [36]. An important aspect of quality improvement is accurate fault diagnosis, and determining types of fault and failures [12, 65]. The timely detection of warranty defects has huge financial implications. Faulty products remaining on the market can cost the automotive industry many millions of dollars in warranty expenses and recall actions [12, 34]. Early detection of warranty defects can significantly reduce this expense and improve customer satisfaction [36]. For several years text mining has been promoted as the solution to warranty issues, particularly the issue of quickly identifying problem areas and fixing them. Instead of technicians trying to select from hundreds of warranty categories or clerks trying to guess which category a technician's written assessment belongs in, text mining software is supposed to "read" written assessments and devise a list of the top warranty areas [8]. Ford Motor Company has invested in text mining technology to extract information contained in warranty claims (and other text sources). While these records contain valuable structured data fields such as vehicle mileage, part codes, and labour operation codes, text mining fills in the gaps between these codes. The results can be dramatic, with some savings estimated in the tens of millions of dollars [36].

Another case in point is the National Highway Traffic Safety Agency in the United States. Here, customer complaints regarding safety related vehicle defects and crashes are collected and investigated. When a certain number of complaints on particular issues have been received, the agency needs to investigate whether this is a significant trend due to some underlying product problem. Defects must be repaired free of charge to the customer and the agency is empowered to ask, and if need be order, the manufacturer to conduct a recall, if warranted by the problem. The data gathered are automobile related information as well as descriptive text information on damages and accidents. Manufacturers are usually motivated to uncover potential problems as soon as possible, making this a candidate for automated problem solving support, in this case the application of data and text mining [34].

To broaden our understanding of how text mining can overcome a variety of problems in manufacturing; we consider, in the following subsection, an important activity within manufacturing companies which is the product development process.

Text Mining within the Product Development Process

In the mid-1980s there was a growing awareness in American companies that the performances of their *Product Development Processes* (*PDP*)*s* were not matching up to that of their Japanese competitors. Japanese manufacturers were able to produce many products much faster and with better quality. Many studies sponsored by the USA government to research the success of the Japanese and seek for new product development methods [174]. This example shows how today's competitive market has created a highly challenging environment for product development. Companies are under increasing pressure to sustain their competitive advantages by reducing product development time and cost while maintaining a high level of quality [21]. Companies that are operating in this competitive market have to plan their product development activities carefully so that they can provide products that achieve customer satisfaction as well as gain sufficient profit and maintain their market position [174]. A good product development process is a key to creating a successful product. A well-organized and coherent development process serves to ensure the efficient delivery of a final product that suits customer's wants. Such products are truly the 'lifeblood' of a company's long term economic existence. Thus, it is no surprise that companies are willing to invest both time and effort to ensure a proper product development process so as to deliver competitive products [98]. The increasing competition amongst companies as well as uncertainty that we are experiencing today, have brought about important trends that need to be considered for the development of new products. These trends, which can be outlined as follows [156]:

- Increasing (technical) product complexity.
- Increasing complexity of the business processes.
- Changing customer expectations/requirements.
- Shorter development time.

have given rise to an increase in the number of unexpected events within the *PDP*. Traditional tools are only partially adequate (either insufficient coverage or simply too late) to cover these unexpected events. As such, new tools are being sought to complement traditional ones [157]. Here, we focus our attention on the use of one such tool, textual data mining. Many different definitions of product development exist. For our purposes, we adopt the following definition provided by de Graaf [55, 103]: *"Product development is a sequence of design processes that converts generally specified market needs or ideas into detailed information for satisfactorily manufacturing products, through the application of scientific, technical and creative principles, acknowledging the requirements set by succeeding life cycle processes".*
The above definition well suits our needs as it recognizes the importance of information and its flow, obtained and enabled by application of technical methodologies such as text mining, in the production of high quality products. Although every organization may follow a slightly different process, the basic elements are usually the same. In essence, the major steps that would usually be incorporated into the *PDP* are [98]:

- Market Need Identification.
- Planning.
- Design.
- Testing and Refinement.
- Production Ramp-up.
- Service and Support.

Table 3.1 provides a summary of a work done within the *PDP* in relation to *DM*. As can be seen from the surveyed literature thus far, many of the studies carried out have focused on numerical databases within the *PDP*. Comparatively less work has been done to explore the potential of textual databases within the *PDP*, although the potential for such data is large and the need for it is immediate [103].

Table 3.1: Summary of DM applications within the PDP [103].

	Market Need Identification	**Planning**	**Design & Testing**	**Production Ramp-up**	**Service & Support**
Numeric Data		Component and part requirement forecasting Changes in generating process plan due to DM Effective configurations for cellular manufacturing systems	Evaluating level of support to design activities Classification of *PDM* and geometry data Estimating engineering properties of technical object/process Knowledge acquisition in engineering design Difficulties of developing a data mining-based engineering support system Determining effect of design parameters for drop test Selecting starting prototype	Specification of chip location for defect measurement Detecting failures for die-level functional testing Identifying second-order process control parameter in magnetic recording facility Quality control for liquid crystal display fabrication Quantifying relationship between balance and vibration test in assembly of turbine motors Refinement of cleaning process in wafer fabrication Identifying critical poor-yield factors in wafer fabrication Identifying the correlation between yield and various wafer parametrical data	Discovery of crucial repair cases from a field service database Field Failure analysis of avionics units Extracting information on expected cost of various service requests
Textual Data	Customer requirements with varied socio-cultural background Extracting information from open ended survey		Identifying shared understanding in design documentation Search and retrieval of design information Relationship among design concepts		Mining customer service database for online machine fault diagnosis

A group of researcher studied 4 textual databases. They were; the Service Centre, the Call Centre, the Problem Response System and the Customer Survey databases. The different phases of the *PDP* in which these databases were found are presented in Figure 3.3.

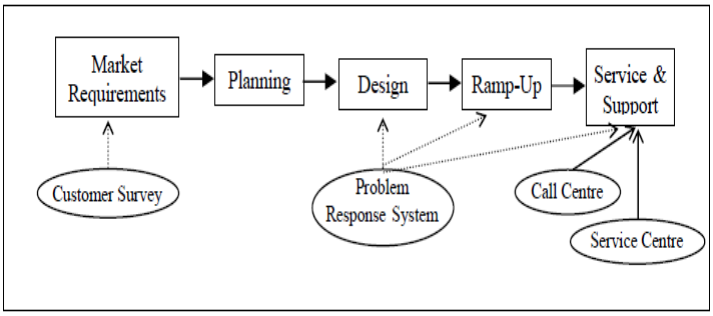

Figure 3.3: Databases studied within the Product Development Process [156].

In the research, it was found that numeric indicators had a low probability of finding the root-causes of the problem and the potential of finding such root-causes was enhanced with the analysis of texts using data mining [156]. The group also concluded that data mining did not only allow for quick feedback but could also provide valuable insights into the data. That insight leaded to better product understanding which was very vital to improving product quality and reliability [156].

3.1.2 A Quality Improvement Tool in Service Sector

In economics and marketing, services are the non-material equivalent of goods [165]. Since most modern products (goods and services) are a combination of intangible acts and tangible goods, some authors see that the above definition is inadequate and to provide a satisfactory definition they characterise services with the following important features [117, 165]:

- *Intangibility*: services cannot generally be seen, tasted, felt, heard or smelled before they are bought.
- *Inseparability*: services are produced and consumed at the same time.

- *Variability*: the quality of the same service may vary depending on who provides it as well as when and how it is provided.
- *Perishability*: services cannot be stored for later sales or use; unsold service time is "lost", that is, it cannot be regained.

In a narrower sense, service refers to quality of customer service: the measured appropriateness of assistance and support provided to a customer. This particular usage occurs frequently in retailing [165]. Due to the heterogeneity in services it is difficult to classify them in a useful manner. However, the classification developed by Lovelock (Table 3.2) appears useful because it provides answer to the question "Why is service quality important?" [117].

Table 3.2: Classification of services by Lovelock [117].

Types of Services	**Examples**
Tangible actions directed at people's bodies.	Health care Passengers transportation Hotels and restaurants Beauty care
Tangible actions directed at goods and other physical possessions.	Freight transport Industrial equipment Repair and maintenance Janitorial services Laundry and dry cleaning Veterinary care
Intangible actions directed at people's minds.	Education Broadcasting Information services Theatres Museums
Intangible actions directed at intangible assets.	Banking Legal services Accounting Securities Insurance

The service sector (or the service industry) is also referred to as the *non-manufacturing sector*. However, there is some disagreement among policy makers and economists as to a precise definition of what constitutes the service or service-

providing sector [51]. The service sector, also known as the *tertiary sector* of industry, is one of the three main industrial categories of a developed economy, the others being the secondary industry (manufacturing), and primary industry (extraction such as mining, agriculture and fishing) [166]. In recent years the service sector has come to be viewed as a dynamic component of the economy, characterized by the large consumption of new technologies and human capital [51]. In 2005, USA was the largest producer of services followed by Japan and Germany (reports the International Monetary Fund) [165]. Economies tend to follow a developmental progression that takes them from a heavy reliance on agriculture and mining, toward the development of industry (e.g., automobiles, textiles, shipbuilding, steel) and finally toward a more service based structure. Whereas the first economy to follow this path in the modern world was the United Kingdom, other economies have later made the transition to service-based, sometimes called post-industrial [166], and Turkish economy is taking its way on this path [41]. We have mentioned how and why it is important to improve quality in this vital sector of the economy. Therefrom the significance of using any tool, such as text mining, that aids this target. The following examples present the quality improvement in some services done by means of text mining:

The Hospital of Walcheren Improves the Quality of Services

The Hospital of Walcheren in the Netherlands serves nearly 120,000 patients each year. In order to comply with a new healthcare quality law, the hospital needed to evaluate the quality of its services based on patient feedback. Due to the number of patients involved, it was important to have an efficient way to track and measure service quality and perform complex analysis on the results. The hospital made use of text mining to analyze the survey results. The quality manager of the hospital reported that text mining had really helped the hospital to identify areas of patient care that needed improvement, and had already contributed significantly to patient well-being and satisfaction [147].

Automated Help Desk

Text mining can automatically identify messages coming from customers via email or WWW forms, categorize them, and route them for an appropriate action. This will not only reduce the cost of customer care, but also increase the quality of customer care by enabling timely and individualized care. For instance, messages that are determined to be complaints can be sent to the customer service department for timely handling. Sales related inquiries could be answered automatically with appropriate product information or sent to the sales manager if it is an important strategic sales lead [146].

Financial Services

The Principal Financial Group implemented a document management and presentation application that allows company employees to use their Web browsers to access current customer statements on its intranet. The text mining tools provide the company with a high-performance document storage, retrieval, and presentation solution. As a result, customers get questions answered immediately because finding the right statement is fast and convenient for service representatives [126].

3.2 Experimental Study

As organizations move from not being able to find information to being overwhelmed by it, it has become evident that text analytics and visualization tools are required to handle the information overflow. These tools perform the following essential tasks [46]:

- Monitor large amounts of information.
- Merge multiple sources of information.
- Reduce the mountain into understandable and analyzable units.
- Measure trends, opinions, competitors, marketing campaigns, and products in the media.

- Compare and track – over time, between products, among competitors, and between marketing messages.
- Discover the unexpected – issues, problems, competitors, or sales.
- Present new business opportunities.
- Visualize the information so that it's quick to assimilate.
- Make information available to roving professionals who are not tied to a location or device.

Given the broad area that text mining applications cover, it is more than expected to exist a fairly rich variety of software supporting text analysis task within different disciplinary context in considerably different ways. In fact, there are software tools that are qualitative-oriented as well as quantitative-oriented tools [145]. Several text mining methods exist, such as vector space model, latent semantic indexing, computational linguistics, and classification. Examples of application areas are intelligent searching, information filtering and routing, text categorization and automatic indexing, while known tools for text mining include WordStat, Oracle Text, IBM Intelligent Miner and Megaputer Text Analyst [159].

In this thesis, based on actual projects, two case studies are described using the text-based software WordStat. Besides, the definition and overview of concepts, operations and techniques used are provided.

WordStat is a text analysis module specifically designed to study textual information such as responses to open-ended questions, interviews, titles, emails, memos, contracts, government reports, journal articles, public speeches, electronic communications, etc. [127, 159]. It may be used for automatic categorization of text using a dictionary approach or various text mining methods as well as for manual coding [127]. The software includes numerous exploratory data analysis and graphical tools that may be used to explore the relationship between the content of documents and information stored in categorical or numeric variables such as the gender or the age of the respondent, year of publication, etc. Relationships among words or categories as well as document similarity may be identified using hierarchical clustering and multidimensional scaling analysis. Correspondence analysis and heatmap plots may be used to explore relationship between keywords

and different groups of individuals. WordStat is not a standalone application but a module that must be run from Simstat, a statistical data analysis software, or QDA Miner, a text management and qualitative coding software [122].

3.2.1 Keyword-In-Context

The Keyword-In-Context (*KWIC*) technique allows one to display in a table the occurrences of either a specific word, or of all words related to a category, with the textual environment in which they occur. The text is aligned so that all keywords appear aligned in the middle of the table. This technique is useful to assess the consistency (or lack of consistency) of meanings associated with a word, word pattern or category [127, 159]. The *KWIC* technique is also useful to highlight syntactical or semantic differences in word usage between individuals or subgroup of individuals. For example, candidates from two different political parties may use the word "rights" in their discourses at the same relative frequency, but we may find that these two groups use this word with quite different meanings [127]. WordStat uses the following in completing its categorization process to determine the similarity of keywords and categories: Jaccard's coefficient, Sorensen's coefficient, Ochiai's coefficient and Cosine theta [159]:

Jaccard's Coefficient

This coefficient is computed from a fourfold table as $a/(a+b+c)$ where a represents cases where both items occur, and b and c represent cases where one item is found but not the other. In this coefficient equal weight is given to matches and non matches.

Sorensen's Coefficient

This coefficient (also know as the Dice coefficient) is similar to Jaccard's but matches are weighted double. Its computing formula is $2a/(2a+b+c)$ where a

represents cases where both items occur, and b and c represent cases where one item is present but the other one is absent.

Ochiai's Coefficient

This index is the binary form of the cosine measure. Its computing formula is $\sqrt{a^2/((a+b)(a+c))}$ where a represents cases where both items occur, and b and c represent cases where one item is present but not the other one. The last coefficient takes into account not only the presence of a word or category in a case, but also how often it appears in this case.

Cosine Theta

This coefficient measures the cosine of the angle between two vectors of values. It ranges from -1 to +1.

3.2.2 Hierarchical Clustering and Multidimensional Scaling

WordStat allows one to further develop categorization by providing various graphic tools to assist the identification of related words or categories. Those tools are obtained by the application of hierarchical cluster analysis and multidimensional scaling on all included words or categories and are displayed in the form of dendrograms and concept maps [127].

3.2.2.1 Multidimensional Scaling Analysis

Multidimensional scaling (*MDS*) has no background theory: it is an exploratory tool for suggesting relationships in data rather than testing pre-chosen hypotheses. There is no agreed criterion which tells us if the scaling is successful, although there are generally accepted guidelines [105]. *MDS* methods seek to represent data points in a lower dimensional space while preserving, as far as is possible, the distances between the data points [64]. Many *MDS* methods exist, differing in how they define the distances that are being preserved, the distances they map to, and how the

calculations are performed [64]. Since *MDS* methods seek to preserve interpoint distances, such distances can serve as the starting point for an analysis. That is, we do not need to know any measured values of variables for the objects being analyzed, only how similar the objects are, in terms of some distance measure [64]. The only assumption that is made by *MDS* is that there is a monotonic relationship between the original and projected pair-wise distances (i.e., if we plot the original distance d_{ij} and the projected distance d'_{ij} of all objects in a two-dimensional diagram, then the points should lie on a smooth line or curve that behaves monotonically) [11, 30]. The algorithm attempts to map each object into a *k*-dimensional space, so that badness of fit function (usually called *stress* or *S*-function in the context of *MDS*):

$$stress = \sqrt{\frac{\sum_i \sum_j (d'_{i,j} - d_{i,j})^2}{\sum_i \sum_j d_{i,j}^2}} \tag{3.1}$$

is minimized, where d_{ij} is the original distance between objects *i, j*=1,2,…,*n* and d'_{ij} is the Euclidean distance between the *k*-dimensional points to which these objects are mapped [1, 30, 160]. Typical algorithms for optimizing this function work iteratively by picking an initial assignment and then trying to improve the solution by picking a point and moving it to the location that would minimize the stress function [30].

A simplified view of the algorithm is as follows [11, 160]:

1. Assign points to arbitrary coordinates in *k*-dimensional space.
2. Compute Euclidean distances among all pairs of points, to form the $\hat{D}$ matrix.
3. Compare the $\hat{D}$ matrix with the input *D* matrix by evaluating the stress function. The smaller the value, the greater the correspondence between the two.
4. Adjust coordinates of each point in the direction that the stress function is minimized.
5. Repeat steps 2 through 4 until stress won't get any lower.

3.2.2.2 Clustering Task

Clustering refers to the grouping of records, observations, or cases into classes of similar objects. A *cluster* is a collection of records that are similar to one another and dissimilar to records in other clusters [90]. Traditionally clustering techniques are broadly divided in hierarchical and partitioning. Hierarchical clustering is further subdivided into agglomerative and divisive [9].

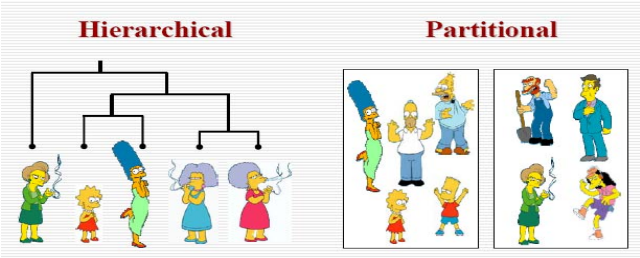

Figure 3.4: Tow types of clustering [162].

The agglomerative are the more important and widely used of the two [64], and this is the method implemented in WordStat. However, in case of large databases, this method is not very practical since it scales at least quadratically with the number of data objects, that is, $O(N^2)$ [106]. The divisive methods start with all of the observations in one cluster and then proceeds to split (partition) them into smaller clusters. The agglomerative methods begin with each observation being considered as separate clusters and then proceeds to combine them until all observations belong to one cluster [40]. However, divisive methods are not generally available, and rarely have been applied [125]. Hierarchical methods of cluster analysis permit a convenient graphical display, in which the entire sequence of merging (or splitting) of clusters is shown. Because of its tree-like nature, such a display is called a *dendrogram* [64].

An example of such a dendrogram is given below Figure 3.5:

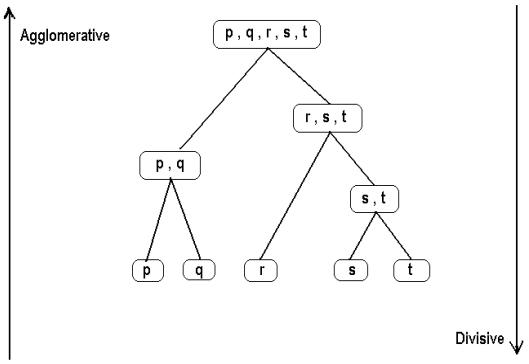

Figure 3.5: The divisive and the agglomerative methods [168].

A matrix of similarity or dissimilarity is provided between every pattern pair [106]. Agglomerative algorithm has several variations depending on the metric used to measure the distances among the clusters. The Euclidean distance is usually used for individual points [50]. Among the most used variations of the hierarchical clustering based on different distance measures are average linkage, complete linkage, single linkage and Ward's linkage [40, 50].

Average Linkage Clustering

Here the distance between two clusters is defined as the average of distances between all pairs of objects, where each pair is made up of one object from each group. Thus, the distance between two clusters r and s, $D(r,s)$, is computed as:

$$D(r,s) = \frac{T_{rs}}{(N_r \times N_s)}$$

$$= \frac{1}{(N_r \times N_s)} \times \sum_{i=1}^{N_r} \sum_{j=1}^{N_s} dis(x_{ri}, x_{sj}), \qquad (3.2)$$

where T_{rs} is the sum of all pair-wise distances between cluster r and cluster s, N_r and N_s are the sizes of the clusters r and s, respectively, $x_{ri} \in r$ and $x_{sj} \in s$, and $dis(x_{ri}, x_{sj})$ is any distance or (dis)similarity measure between x_{ri} and x_{sj}. At each stage of hierarchical clustering, the clusters r and s, for which $D(r,s)$ is the minimum, are merged [133]. The figure below (Figure 3.6) illustrates average linkage clustering:

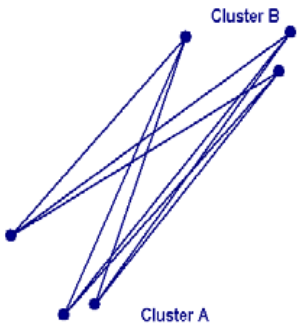

Figure 3.6: Average linkage clustering [168].

Hierarchic Agglomerative Clustering Algorithm

Given a set of N items to be clustered, and an $N \times N$ distance (or similarity) matrix, the basic process of hierarchical agglomerative clustering is this [125]:

1. Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain.
2. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.
3. Compute distances (similarities) between the new cluster and each of the old clusters.
4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

Step 3 can be done in different ways, which is what distinguishes *single-linkage* from *complete-linkage* and *average-linkage* clustering.

3.2.3 Dendrogram

WordStat uses an average-linkage hierarchical clustering method to create clusters from a similarity matrix. The result is presented in the form of a dendrogram (see Figure 3.7), also known as a tree graph. In such a graph, the vertical axis is made up of the items and the horizontal axis represents the clusters formed at each step of the

clustering procedure [127]. Words or categories that tend to appear together are combined at an early stage while those that are independent from one another or those that don't appear together tend to be combined at the end of the agglomeration process [127].

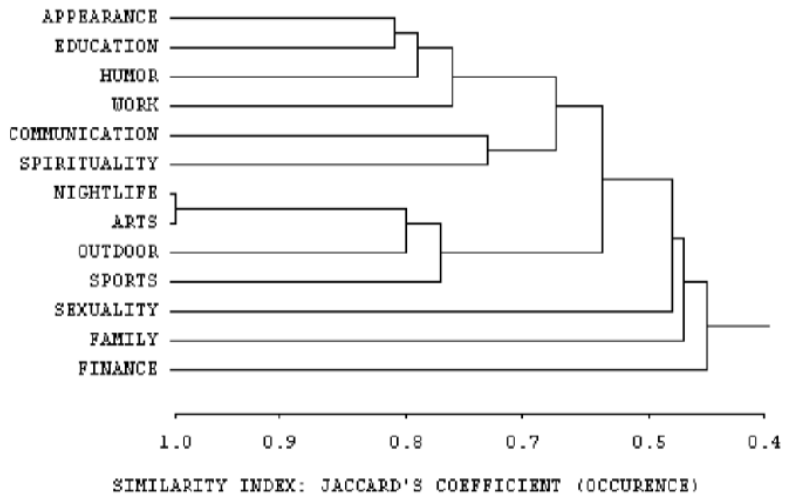

Figure 3.7: A dendrogram [127].

3.2.4 Proximity Plot

The proximity plot is the most accurate way to graphically represent the distance between objects by displaying the measured distance from a selected object to all other objects on a single axis. It is not a data reduction technique but a visualization tool to help one extract information from the huge amount of data stored in the distance matrix at the origin of the dendrogram and the multidimensional scaling plots. In this plot, all measured distances are represented by the distance from the left edge of the plot. The closer an object is to the selected one, the closer it will be to the left [127].

3.2.5 Comment Card Analysis

Customer satisfaction represents a modern approach for quality in enterprises and organizations [91, 101]. Accurate and timely competitor and customer intelligence

enhances hotel effectiveness and customer satisfaction. In the era of information explosion, hospitality practitioners are overloaded by data. They often complain that they have too much data but they do not have enough understanding. Hoteliers have long recognized the benefits of installing *information technology* (*IT*) systems. However, since a majority of business information exists in the form of unstructured or semi-structured text documents, such as those stored in a hotel's internal databases or in Web-based data sources, and the traditional way of processing text information requires substantial investment of money, time, and human resources, the need for a means of textual information management, such as text mining, is pressing. Text mining can analyze the huge textual information that can be found in a hotel's internal databases and external sources [91].

In the following subsections the analysis of a large volume of comment cards [102] about hotel services of a major hospitality organization, using the so called WordStat with *QDA* Miner softwares, is illustrated.

3.2.5.1 Understanding the Case

The organization wishes to identify key issues and problems. This will allow the organization to improve its service and customer satisfaction. Reading through each of the comments manually is quickly recognized as a time consuming and costly process. The data consists of a few thousand individual comment cards that were returned to the hotel when residents checked out of their rooms [102]. "These are real comments at a real chain of hotels from somewhere in Europe, but the data has been disguised. The Hotel names are fictitious, and all identification information has been removed" [102]. The comments are coded under 'room', 'housekeeping', 'staff', 'service', 'food', 'reception', 'general', and 'other' by the staff of each hotel when collecting the cards. Some customers preferred not to fill in the cards with their ages or genders or the both. Customers are grouped according to their ages into four categories (20-30), (31-45), (46-60), (61+) in addition to the 'unknown' category. In this experimental study, our main focus would not be on getting perfect results or making detailed analysis, although trying to obtain the best possible ones, but rather build up a picture or roadmap of feasible textual data management by applying

various text mining techniques and approaches using qualitative data analysis software package.

3.2.5.2 Data Preprocessing

First step after we input the text documents in the system is the preprocessing step. In this step the following stages of preprocessing was carried out:

- Filtering or exclusion of stopwords.
- Applying Porter's stemming algorithm.
- Removal of words that occurred no more than 5 times and no less than 2 times, in the entire corpus.

3.2.5.3 Basic Understanding

Let us take a look at the following charts:

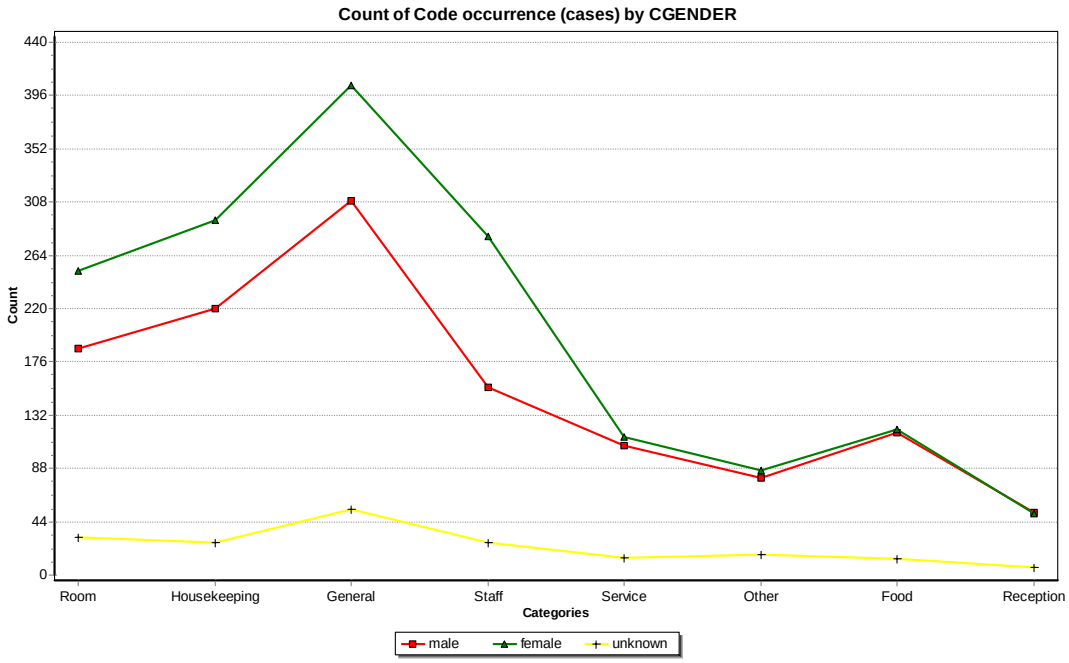

Figure 3.8: Count of code occurrence (cases) by gender.

Figure 3.8 shows that, there are more females than males that filled out cards.

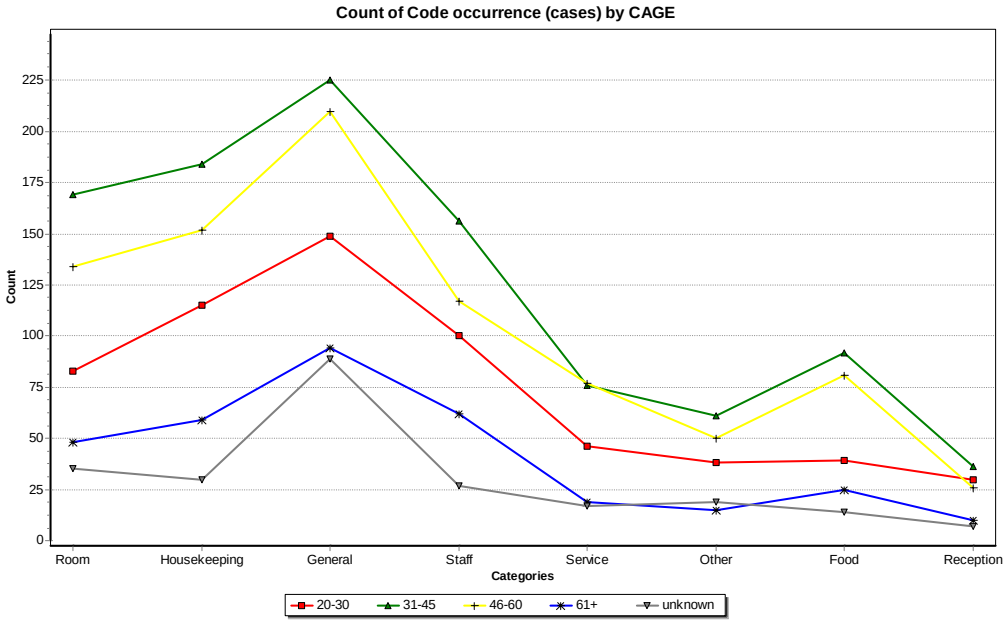

Figure 3.9: Count of code occurrence (cases) by age.

Figure 3.9 shows that, most responders are between 31 and 45 (then between 46 and 60) years of age.

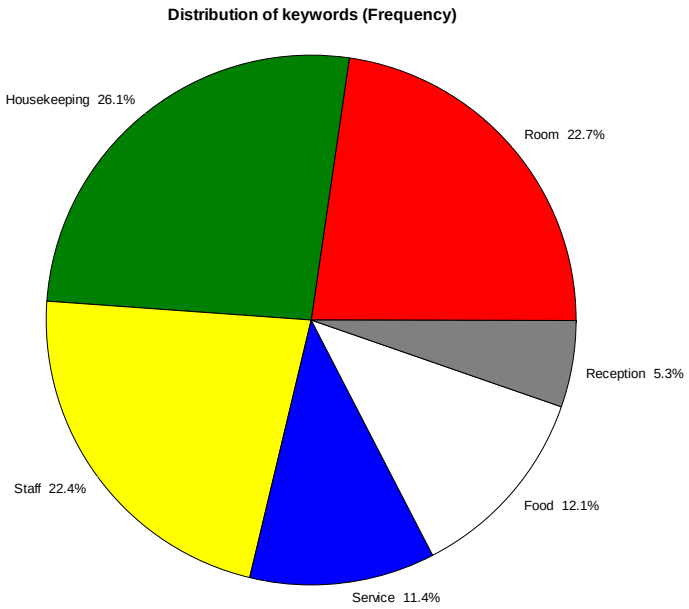

Figure 3.10: Distribution of keywords (frequency).

Figure 3.10 shows that, most comments are about the housekeeping and the room.

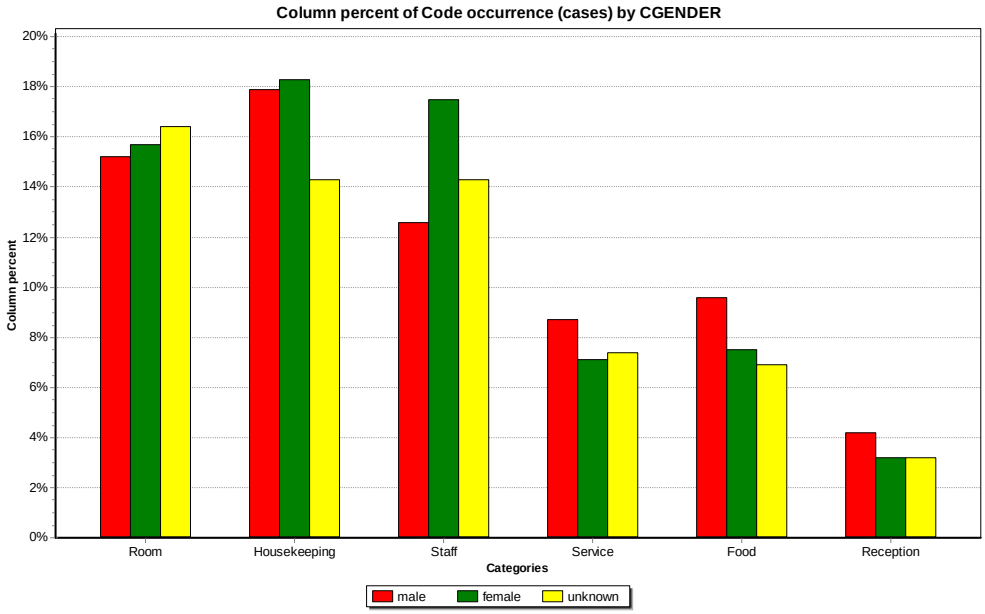

Figure 3.11.a: Column percent of code occurrence (cases) by gender.

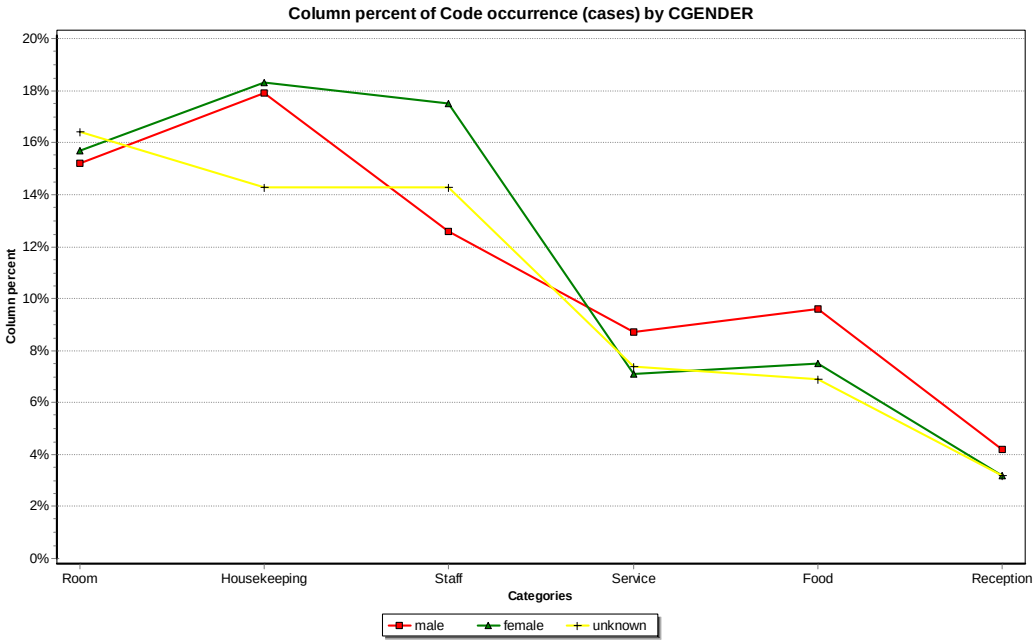

Figure 3.11.b: Column percent of code occurrence (cases) by gender.

Figures 3.11.a and 3.11.b show that, males talk about food, service, and reception, while females talk about the staff, room and housekeeping.

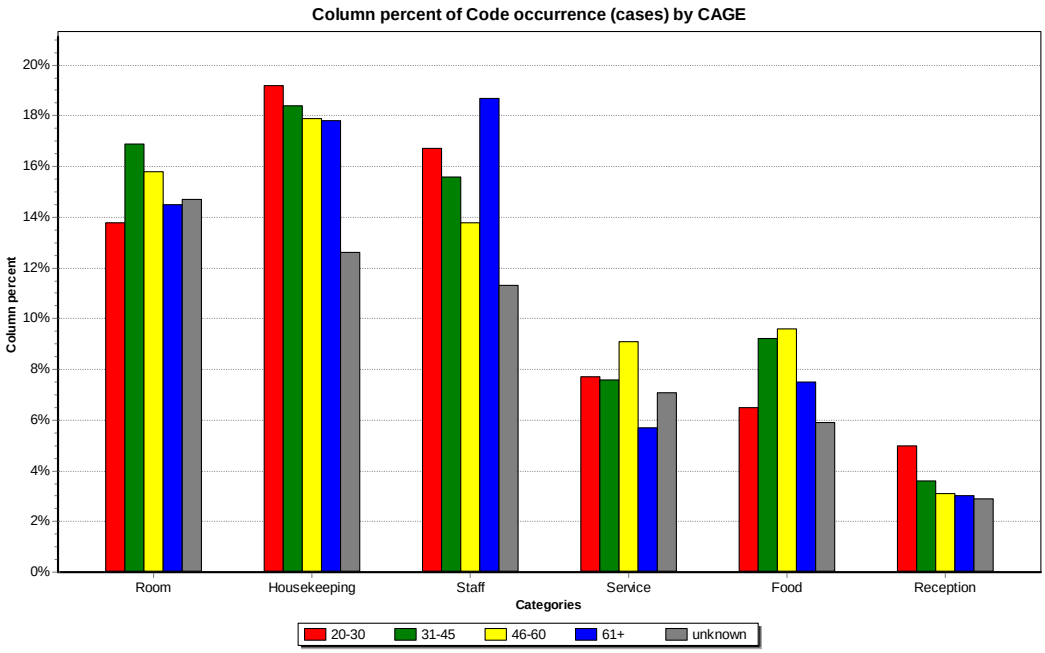

Figure 3.12.a: Column percent of code occurrence (cases) by age.

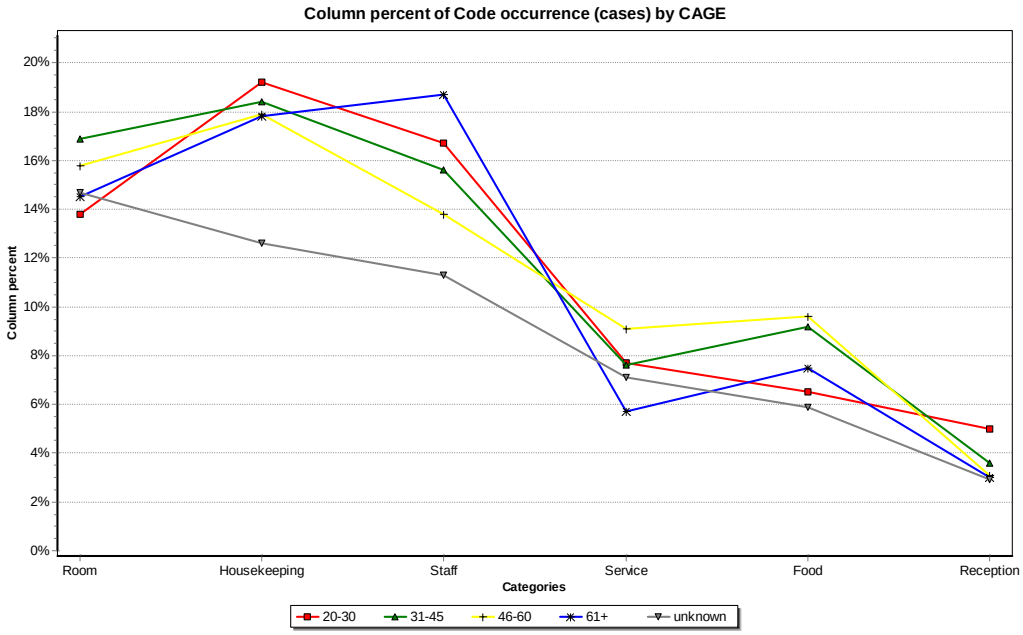

Figure 3.12.b: Column percent of code occurrence (cases) by gender.

Figures 3.12.a and 3.12.b show that, (20-30) year olds discuss housekeeping and reception, (31-45) year olds discuss the room, (46-60) year olds discuss the service and the food, and (60+) year olds discuss the staff.

The above-mentioned coding procedure can be done automatically by applying the automatic text categorization. For this purpose, we may consider Naïve Bayes Classifier or *k-NN* method. Choosing k = 1, 5, 30 and examining the performance of *k-NN* and Naïve Bayes Classifier on the comment cards, as displayed below (Figures 3.13, 3.14, and 3.15), we found that 1-*NN* has shown the best performance.

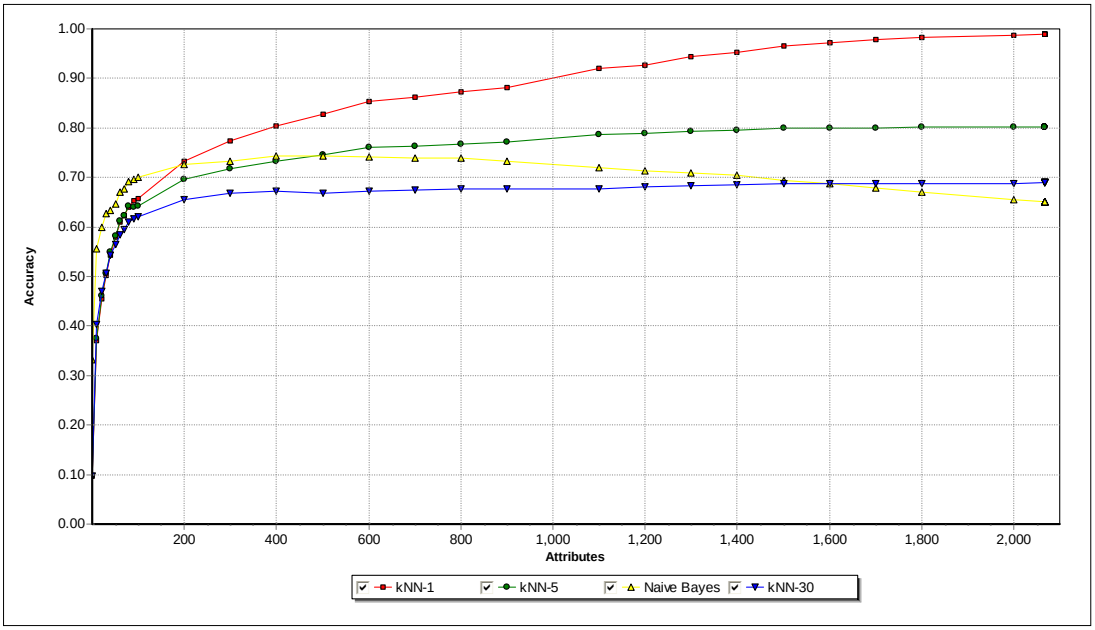

Figure 3.13: The performance of k-NN with different values of k and NB.

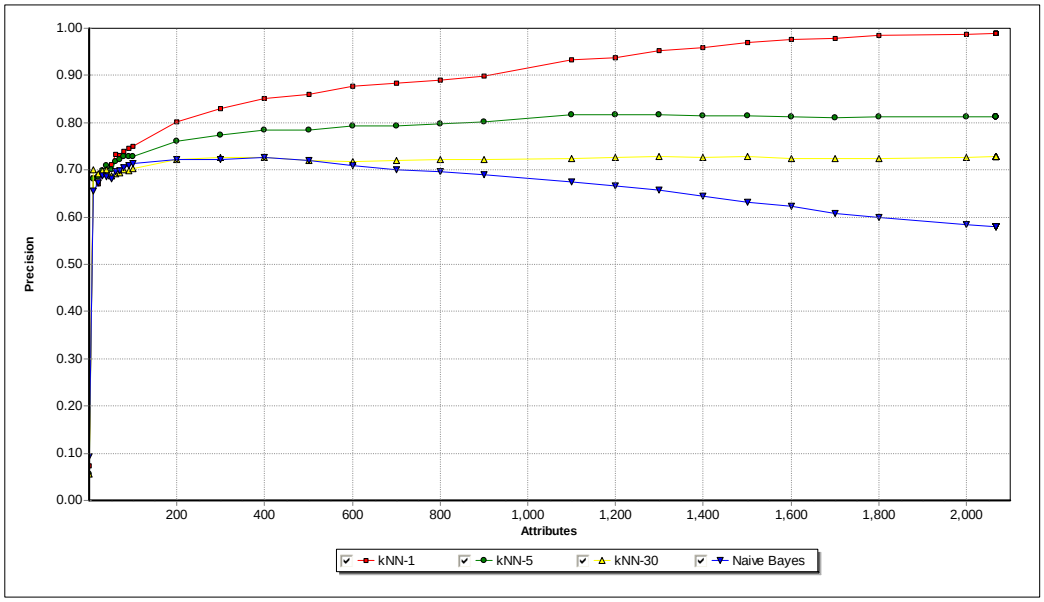

Figure 3.14: The precision scores.

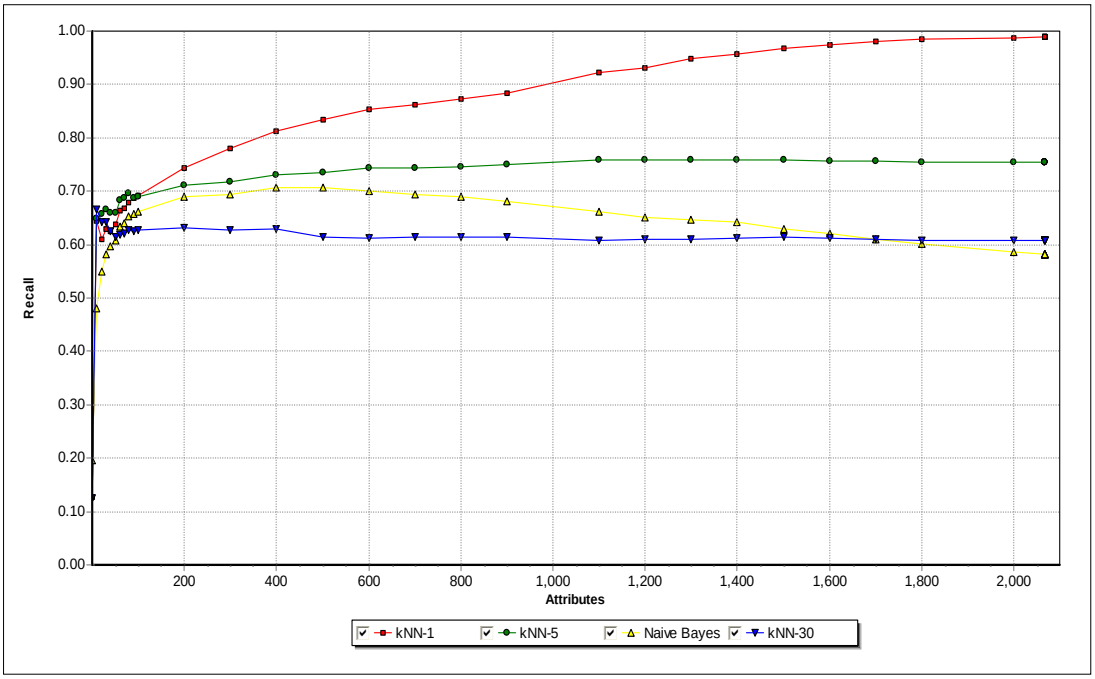

Figure 3.15: The recall scores.

The difference between the manual coding or categorization and the result of the automatic one, implementing 1-*NN*, is illustrated in Table 3.3, in which the codes under 'actual' are those their coding done manually and the ones under 'predicted' stand for the 1-*NN* categorization results.

Table 3.3: The manual coding (actual) and automatic categorization (*predicted*).

Record #	Actual	Predicted
2	Room	General
13	Housekeeping	Room
37	General	Service
103	Room	General
138	Reception	General
347	Room	General
364	Room	General
489	Service	Other
715	Other	Housekeeping
836	Room	General
956	Reception	Staff
976	Food	Staff
1054	Staff	Reception
1063	General	Room
1170	Staff	General
1220	General	Other
1406	General	Other
1485	General	Service
1611	Room	General
2077	General	Other
2238	General	Room
2270	Room	General
2289	Service	General
2531	General	Room
2533	Room	General
2541	Room	General
2646	General	Staff
2767	Room	General
2838	Room	General
2898	Room	General
2975	Other	General

Slightly different occlusions obtained, here, from those we have already acquired making use of the manually coded comments.

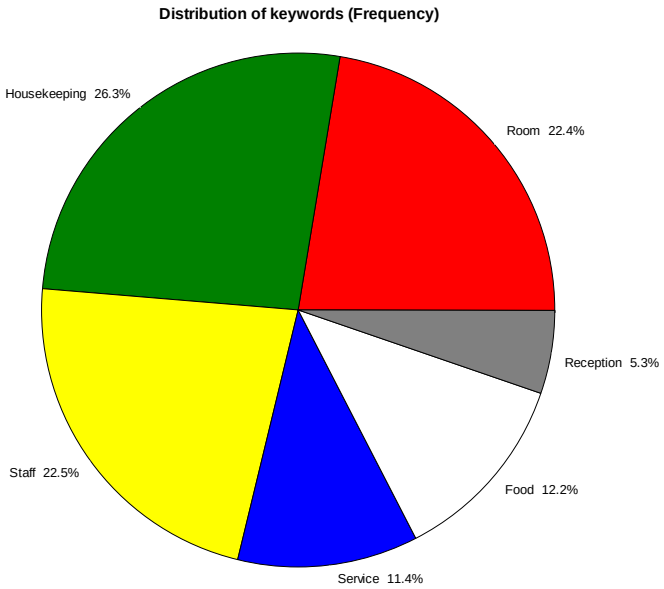

Figure 3.16: Distribution of keywords (*frequency*).

Figure 3.16 shows that, most comments are about the housekeeping and the staff. Here the percentage of comments coded with staff is a little higher than that of room, while in manually coding, the situation was in reverse. However, by examining the following two charts (Figures 3.17 and 3.18), conclusions similar to those obtained previously can again be stated.

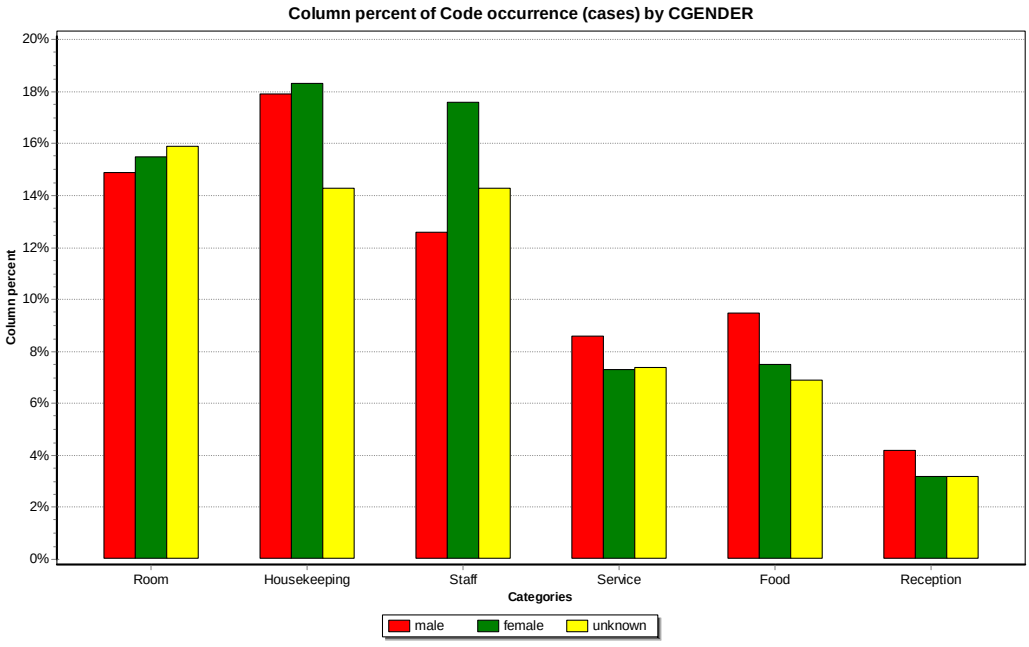

Figure 3.17: Column percent of code occurrence (*cases*) *by gender.*

Figure 3.17 shows that, males talk about food, service, and reception, while females talk about the staff, room and housekeeping.

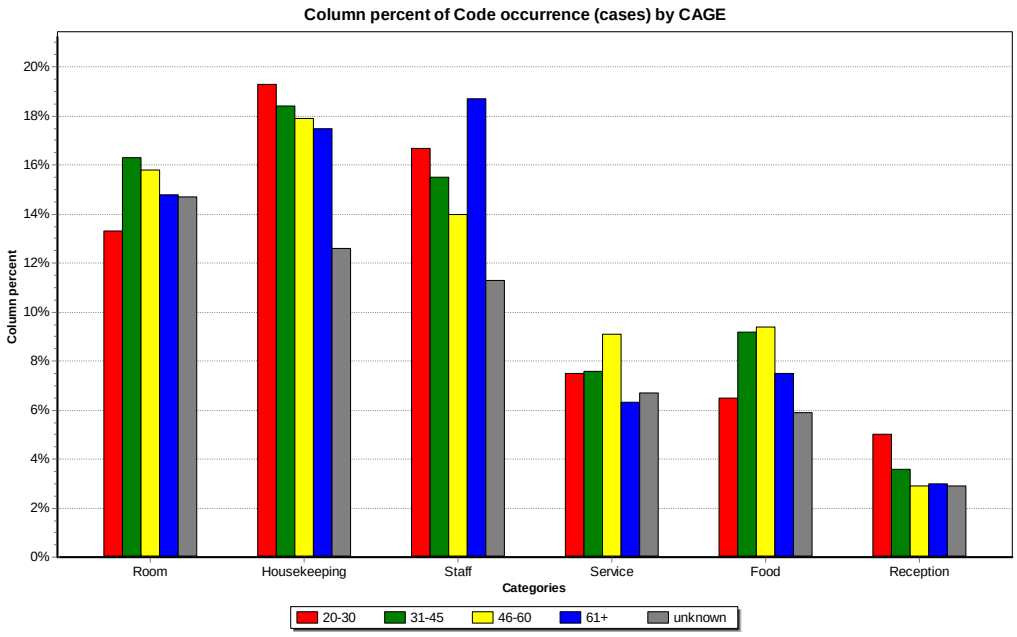

Figure 3.18: Column percent of code occurrence (*cases*) *by age.*

Figure 3.18 shows that, (20-30) year olds discuss housekeeping and reception, (31-45) year olds discuss the room, (46-60) year olds discuss the service and the food, and (60+) year olds discuss the staff.

Automatic text categorization can be done with lower cost, and greater consistency than the manually categorization. It maybe done with greater accuracy (or, maybe not). However, while human error-rate is often ignored, computer error-rate is often discussed [15].

3.2.5.4 Exploring Some of the Interesting Correlations

Applying techniques like *KWIC* (text-base navigation), entity extraction, and search and retrieve, we can explore some interesting correlations and facts that spotlight on essential matters and problems. This will help the organization to improve its service and customer satisfaction. From the entity extraction list tabulated by the significance of the extracted keywords from the comments, measured by *Tfidf*, and

utilizing the above mentioned techniques, we can examine main issues relating the hotels. Table 3.4 displays part of the entity extraction list, and Table 3.5 consists of the extracted keywords to be chosen for exploration.

Table 3.4: Part of the entity extraction list.

	TF • IDF
AIR CONDITIONING	125
FRIENDLY STAFF	78
ROOM SERVICE	71.2
FRIENDLY AND HELPFUL	69.2
EXCELLENT SERVICE	66.2
ENJOYABLE STAY	64.7
STAFF VERY FRIENDLY	56.9
FRONT DESK	52
GREAT STAY	50.4
HELPFUL STAFF	50.4
PLEASANT STAY	48.7
ENJOYED OUR STAY	41.8
CAR PARK	40

Table 3.5: Chosen keywords.

	TF • IDF
AIR CONDITIONING	125
FRIENDLY STAFF	78
TOWELS IN ROOM	9

In order to investigate and have insights into the keywords in Table 3.4, *KWIC* (text-base navigation) technique (Tables 3.6 and 3.7) and search and retrieve technique (Table 3.8) were utilized. The latter technique was also used to further explore, within the comments, important and remarkable issues extracted, after specifying proper filtering criteria (Tables 3.9 and 3.10).

Table 3.6: Part of KWIC table of 'air'.

KEYWORD			CHOTEL
Hotel need refurbish. No	air	condition , very noisi at front. l	Ambassador City
ery room should at least have a fan or	air	condition.	Ambassador City
	Air	condition? Abil to cope with a	Ambassador City
Telli broken,	air	con.too loud. Two door stiff, w	Best Western
Doubl glaz useless. Nois bad. No	air	condition. Couldn't open windo	Best Western
	Air	con. Stop work at 11pm, disapp	Days Inn
Staff very good. Very noisi	air	con.	Days Inn
Room hot, hotel need more fan or	air	con.	Days Inn
	Air	condition in all room please	Days Inn
	Air	condition malfunc. Woke in mi	Days Inn
Too warm in bedroom.	Air	condition need	Days Inn
Once	air	condition room were receiv, ev	Days Inn
Suggest mak	air	condition standard	Days Inn
Bigger towel and	air	condition	Days Inn
	Air	condition	Days Inn
	Air	condition	Days Inn
All room should be	air	condition	Days Inn
Get	air	condition. The room are very h	Days Inn
Aw look old fashion hairdryer - feebl	air	flow, 7.30am. Disturb from ph	Days Inn
Need more	air	in room	Days Inn
Need	air	con. Room far too hot & windo	Embassy Suites
Too warm. Room need	air	con.	Embassy Suites
Because of the heat wave,	air	cond in bedroom would have b	Embassy Suites
	Air	Cond instal it!	Embassy Suites
Need	air	cond. Room too hot to sleep - I	Embassy Suites
Need	air	condition and room need face l	Embassy Suites
You must get	air	condition even the lowest mote	Embassy Suites
	Air	condition is miss	Embassy Suites
	Air	condition is need urgent	Embassy Suites
Lack of	air	condition made our stai less co	Embassy Suites
	Air	condition need or fan offer.	Embassy Suites
	Air	condition requir	Embassy Suites
Room condition was not satisfactori.	Air	Condition strongli recommend.	Embassy Suites
The fan was appreci however	air	condition would be love.	Embassy Suites
Need	air	condition	Embassy Suites
Room need	air	condition	Embassy Suites
Put in	air	condition	Embassy Suites
Room were hot - no	air	condition	Embassy Suites
	Air	Condition!	Embassy Suites
	Air	condition. Milk gone off. Dubl	Embassy Suites
Room too hot. No	Air	condition.	Embassy Suites
Would not stai in hot weather. Like	air	condition.	Embassy Suites
Room too warm. Couldn't find	air	condition.	Embassy Suites
n was the intens heat in bedroom - no	air	even with window open	Embassy Suites
Room very warm, no wai to move	air	to cool off.	Embassy Suites
	Air	ventil was bad	Embassy Suites

- The Days Inn and Embassy Suites must provide their rooms with air conditioners.

Table 3.7: Part of KWIC table of 'staff'.

1739	Very friendly	staff		female	46-60	Best Western
1788	:cellent service and wonderful	staff		unknown	unknown	Best Western
2318	Excellent service. Friendly	staff		male	20-30	Best Western
2405	Very friendly and helpful	staff		female	31-45	Best Western
2363	sed. Very friendly and helpful	staff	. Excellent food	female	31-45	Best Western
1735	Very efficient	staff	. Hand soap, hair conditi	female	31-45	Best Western
2409	ovely professionalism of your	staff	. Particularly the night s	male	31-45	Best Western
1723	Overall	staff	extremely helpful	female	61+	Best Western
1790	y at hotel, just came for meal.	Staff	friendly and helpful. Ple	female	31-45	Best Western
1748	Didn't appreciate a member of	staff	knocking and then enteri	female	20-30	Best Western
2939		Staff	most helpful. Very frien	female	20-30	Best Western
2367	Well accepted. Very pleasant	staff	overall	male	31-45	Best Western
2382	t sleep was impossible. Room	staff	walked in when I was in	female	20-30	Best Western
2423		Staff	went out of way to be he	female	46-60	Best Western
1370	Fantast friendli	staff	- will be back.	female	31-45	Circus Circus
1348	brilliant	staff		male	unknown	Circus Circus
1363	More good work femal	staff	0	male	20-30	Circus Circus
1366	Servic	staff	and recep were very help	male	31-45	Circus Circus
1362	Overal	staff	and servic were very ver	female	20-30	Circus Circus
1361	Excel	staff	friendli folk.	male	61+	Circus Circus
1357	Restaur	staff	very friendli and help ev	male	31-45	Circus Circus
1359		Staff	very friendli.	female	61+	Circus Circus
1360	Delici food and great servic,	staff	very friendli.	male	20-30	Circus Circus
1352	street (was not stai overnight).	Staff	went out of their wai to l	female	20-30	Circus Circus
1362	ood, but the dinner and servic	staff	were a delight and worth	female	20-30	Circus Circus
1358		Staff	were excel.	female	46-60	Circus Circus
1365		Staff	were excel.	female	61+	Circus Circus
1364	The	staff	were realli nice especi th	female	20-30	Circus Circus

- The staff at, the Best Western and the Circus Circus must be doing their jobs rightly.

Table 3.8: Part of search and retrieve table of 'towel'.

Door of the bedroom was not opening with card. On last night onl	female	46-60	Econo Lodge
A shower cap and body lotion would have been useful. The TOW	female	20-30	Embassy Suites
Room not clean, no TOWELs. Bin full of rubbish. Curtains dirty	unknown	31-45	Embassy Suites
Public ladies was a disgrace - no toilet roll, no TOWELs. A disgra	female	unknown	Embassy Suites
Not enough TOWELs	male	31-45	Embassy Suites
Family booking for 3 - could have done with more TOWELs, brea	female	31-45	Embassy Suites
Rooms too hot and bigger bath & TOWELs required. Lovely frien	female	31-45	Embassy Suites
Bath TOWELs are too small	male	46-60	Embassy Suites
Get bigger waste baskets more TOWELs	female	61+	Embassy Suites
We were happy to see extra bed made up, TOWELs for 3rd persc	female	31-45	Flamingo Las Vegas
Room too hot, no A/C. TOWEL bar does not heat. Bar glassware	female	31-45	Holiday Inn
Holes in TOWELs!	male	20-30	Holiday Inn
I am all for conservation and the environment but one small soap	male	61+	MGM Grand
Need more TOWELs and tissues in bathroom. Need more TV cha	male	61+	MGM Grand
Wardrobe not long enough to hang dresses. Bags for sanitary TOV	female	31-45	Mainstay Suites
More TOWELs need to be put in the room. The water needs to be	female	20-30	Mainstay Suites
Bathroom very old, TOWEL rack broke. Needs refurbishment	female	20-30	Mandalay Bay
TOWELs are not replaced! Rubber mat for bath tub to prevent acc	female	31-45	Mandalay Bay
I don't think the floors were swept. Everyday we were missing on	female	46-60	Mandalay Bay
More information about facilities would be better in the rooms. A	male	20-30	Motor Inn
Room was not cleaned very well on Sunday. Attention to detail pc	female	46-60	Motor Inn
Telly fault reported, but not acknowledged or dealt with. Leisure c	male	unknown	Motor Inn
No TOWELs in room and housekeeping didn't apologize. A bit ab	female	20-30	Motor Inn
No bathTOWELs available in my room. A basic requirement	female	unknown	Motor Inn
Some delay in receiving TOWELs on arrival	male	61+	Motor Inn
Bathroom dirty, no TOWELs	male	20-30	Motor Inn
On arrival, no light bulbs in lamps. One TOWEL missing on day t	female	61+	Motor Inn
Attention to detail is lacking. No extra TOWELs, replacement of	female	20-30	Motor Inn
Positive and friendly staff, good view central location. Very noisy	female	46-60	Quality Inn
Room and TOWELs not smelling fresh. Pleasant staff but not effe	female	46-60	Quality Inn
Needed extra TOWELs for group, but didn't ask	female	31-45	Radisson
We were three sharing room, but on two occasions, we had to req	male	46-60	Radisson
Not enough TOWELs	female	20-30	Radisson
Please upgrade the TOWELs. Andy the Porter was helpful.	male	31-45	Ramada Inn
Need more TOWELs.	female	20-30	Ramada Inn
More TOWELs.	female	20-30	Ramada Inn
Need extra TOWELs.	female	31-45	Ramada Inn
Bigger TOWELs would be a distinct advantage.	male	46-60	Sleep Inn
Clock and radio should be provided in rooms. Need bath TOWEI	female	46-60	Sleep Inn
Consider room service - Some people will pay for this facility. Ba	female	20-30	Sleep Inn
Housekeeping could not deliver TOWELs requested. Phone bill d	female	31-45	The Luxor
Very poor ventilation. Shower temperature varied, kept running v	female	31-45	The Mirage
The camp bed was unacceptable, our third person slept on the floc	female	31-45	The Mirage

- The Motor Inn and Embassy Suites should reinvest in towels.

Table 3.9: Part of search and retrieve table of 'shower'.

Text	CHOTEL
Also SHOWER head leaks badly	Best Western
SHOWER sprayed horizontally and soaked room.	Best Western
SHOWER broken and kettle broken	Best Western
Also SHOWER head control did not working	Best Western
SHOWER gel ran out in bathroom.	Best Western
SHOWER is not ok, only hot water	Best Western
SHOWER unacceptable, like a spray.	Best Western
SHOWER knob very stiff.	Best Western
SHOWER virtually intolerable.	Best Western
SHOWER too hot, not able to use it.	Best Western
The SHOWER in room 102 needs to be fixed.	Best Western
SHOWER spray needs to be changed	Best Western
No tea-cups, no SHOWER gel, bulb blown.	Best Western
Cold water control on SHOWER not working.	Best Western
SHOWER stopped working half way through washing hair - not i	Quality Inn
No soap, SHOWER cap, ashtray or no drawers to put your belong	Quality Inn
Need to fix SHOWER	Quality Inn
SHOWER in room tended to overflow very fast	Quality Inn
Room 528 SHOWER sprays horizontal, needs repair	Quality Inn
I could not take a SHOWER.	Quality Inn
SHOWER cold water tap cannot be opened .	Quality Inn
SHOWER room rather mouldy - recommend bleach and toothbrus	Quality Inn
The room was very poor, mould all around the SHOWER, no toil	Quality Inn
Dinner poor, SHOWER temperature inconsistent.	Quality Inn
SHOWER does not work properly	The Mirage
Poor room ventilation, nconsistant temperature in SHOWER.	The Mirage
SHOWER temperature varied, kept running very hot intermittentl	The Mirage
Inconsistent water temp.when SHOWERing	The Mirage
SHOWER very hot, spurting very hot	The Mirage
SHOWERs have very unpredictable temp..	The Mirage
The SHOWER is really a problem, it keeps shooting to very high	The Mirage
SHOWER and tub water temp. difficult to control	The Mirage
Thermostatically controlled SHOWERs	The Mirage
SHOWER wouldn't work	The Mirage
Hot water in SHOWER fluctuated in temp. and knocking of wate	The Mirage
Hot water in SHOWER goes on and off at times, scalding at time	The Mirage
SHOWER temp.goes normal to very hot periodically, scalding!	The Mirage
SHOWER unclean.	The Mirage
Fix the SHOWER in room.	The Mirage

- Apparently showers in The Mirage, Quality Inn and Best Western do not work properly.

Table 3.10: Part of search and retrieve table of 'washcloth' and 'facecloth'.

Text	CAGE	CHOTEL
Only thing missing was WASHCLOTHs	61+	The Mirage
WASH CLOTHs would be nice	61+	The Mirage
No hand soap or FACE CLOTHs.	61+	Mandalay Bay
One hand towel and now FACECLOTH or tissues seems meanness rather than for ec	61+	MGM Grand
Please add small FACE CLOTHs to linens.	61+	Grand Hyatt
No WASH CLOTHs.	61+	Flamingo Las Vegas
No WASHCLOTHs in room.	61+	Days Inn
Need FACE CLOTHs.	46-60	The Venetian

- People of age (61+) really need their washcloths (facecloths).

The conclusions can continue, but we are going to stop now for the sake of brevity. There are a quite lot of interesting facts to report back to the hotel management.

3.2.6 The Analysis of Miles per Gallon

Data mining can be used in combination with text mining. The miles per gallon (*MPG*) analysis is a very simple example on this. Fuel efficient cars sell better and are nicer to the environment. While this is too simple of an analysis for commercial application, the ideas and techniques used present several insights into how companies might use data mining with text mining in manufacturing. When an analyst in a large automobile company would like to analyze how characteristics of a car influence its miles per gallon ratio (in other words, how does the *MPG* ratio depend on the other characteristics), he (she) might assess how economical a new car should be and how that compares with the actual *MPG* of the created prototype. The model of miles per gallon in relation to other characteristics provides one with an idea of what parameters are most important when one looks for an economic car. Such a rule could provide a rough guidance for designers of a new car who are constrained by certain limits on how economical the future car should be.

The data consists of 398 records of different cars collected from a test performed in the year 1982. For each car the following attributes are provided [102].

- Average miles per gallon
- Number of the engine cylinders
- Cylinder displacement in cubic inches

- Engine horsepower
- Weight in pounds
- Average number of seconds to accelerate to 100 miles per hour
- Year of production
- Country of manufacturer
- Name of model

The data form is as demonstrated below (Table 3.11):

Table 3.11: Part of MPG data table.

Mpg	Cylinders	Displacement	Power	Weight	Acceleration	Year	Origin	Model
18	8	307	130	3504	12	70	USA	Chevrolet Chevelle Malibu
15	8	350	165	3693	11.5	70	USA	Buick Skylark 320
18	8	318	150	3436	11	70	USA	Plymouth Satellite
16	8	304	150	3433	12	70	USA	Amc Rebel Sst
17	8	302	140	3449	10.5	70	USA	Ford Torino
15	8	429	198	4341	10	70	USA	Ford Galaxie 500
14	8	454	220	4354	9	70	USA	Chevrolet Impala
14	8	440	215	4312	8.5	70	USA	Plymouth Fury Iii
14	8	455	225	4425	10	70	USA	Pontiac Catalina
15	8	390	190	3850	8.5	70	USA	Amc Ambassador Dpl
15	8	383	170	3563	10	70	USA	Dodge Challenger Se
14	8	340	160	3609	8	70	USA	Plymouth 'Cuda 340
15	8	400	150	3761	9.5	70	USA	Chevrolet Monte Carlo
14	8	455	225	3086	10	70	USA	Buick Estate Wagon (Sw)
24	4	113	95	2372	15	70	Japan	Toyota Corona Mark Ii
22	6	198	95	2833	15.5	70	USA	Plymouth Duster
18	6	199	97	2774	15.5	70	USA	Amc Hornet
21	6	200	85	2587	16	70	USA	Ford Maverick
27	4	97	88	2130	14.5	70	Japan	Datsun Pl510
26	4	97	46	1835	20.5	70	Europe	Volkswagen 1131 Deluxe Se
25	4	110	87	2672	17.5	70	Europe	Peugeot 504
24	4	107	90	2430	14.5	70	Europe	Audi 100 Ls
25	4	104	95	2375	17.5	70	Europe	Saab 99E
26	4	121	113	2234	12.5	70	Europe	Bmw 2002

Text mining will soon help the analyst to determine ‘fuel efficient’ cars, if we accept they are those having an average miles per gallon higher than or equal to (40), by cross-tabulating the data, making use of the model and origin names (Table 3.12), and accordingly visualize the data for uncomplicated comparisons. This also will aid in the extraction of keywords (models). For further insight, the related information to

each extracted entity then can beneficially be rearranged and sorted according to the desired characteristic, here by MPG (Tables 3.13, 3.14, 3.15, 3.16, 3.17 and 3.18).

Table 3.12: Cross-tabulation table.

	40.8	40.9	41.5	43.1	43.4	44	44.3	44.6	46.6
BMW									
CADILLAC									
CHEVROLET									
COROLLA									
CORONA									
DATSUN	1								
EUROP		1	1	1	1	1	1		
FIAT									
FORD									
HARDTOP									
HONDA								1	
ISUZU									
JAPAN	1							1	1
LECAR		1							
MAZDA									1
MERCED-BENZ									
NISSAN									
PEUGEOT									
RENAULT		1							
TOYOTA									
USA									
VOLKSWAGEN				1					
VOLVO									
VW			1		1	1	1		

Table 3.13: Part of KWIC table of 'Datsun'.

KEYWORD		MPG	CYLINDERS	DISPLACEME	POWER	WEIGHT	ACCELERATI	YEAR	VARIABLE
Datsun	610	22	4	108	94	2379	16.5	73	MODEL
Datsun	810	22	6	146	97	2815	14.5	77	MODEL
Datsun	200-Sx	23.9	4	119	97	2405	14.9	78	MODEL
Datsun	710	24	4	119	97	2545	17	75	MODEL
Datsun	810 Maxima	24.2	6	146	120	2930	13.8	81	MODEL
Datsun	Pl510	27	4	97	88	2130	14.5	70	MODEL
Datsun	Pl510	27	4	97	88	2130	14.5	71	MODEL
Datsun	510	27.2	4	119	97	2300	14.7	78	MODEL
Datsun	510 (Sw)	28	4	97	92	2288	17	72	MODEL
Datsun	B210	31	4	79	67	1950	19	74	MODEL
Datsun	210	31.8	4	85	65	2020	19.2	79	MODEL
Datsun	710	32	4	83	61	2003	19	74	MODEL
Datsun	B-210	32	4	85	70	1990	17	76	MODEL
Datsun	280-Zx	32.7	6	168	132	2910	11.4	80	MODEL
Datsun	200Sx	32.9	4	119	100	2615	14.8	81	MODEL
Datsun	F-10 Hatchback	33.5	4	85	70	1945	16.8	77	MODEL
Datsun	1200	35	4	72	69	1613	18	71	MODEL
Datsun	510 Hatchback	37	4	119	92	2434	15	80	MODEL
Datsun	210 Mpg	37	4	85	65	1975	19.4	81	MODEL
Datsun	310	37.2	4	86	65	2019	16.4	80	MODEL
Datsun	310 Gx	38	4	91	67	1995	16.2	82	MODEL
Datsun	B210 Gx	39.4	4	85	70	2070	18.6	78	MODEL
Datsun	210	40.8	4	85	65	2110	19.2	80	MODEL

Table 3.14: Part of KWIC table of 'Mazda'.

KEYWORD		MPG	CYLINDERS	DISPLACEME	POWER	WEIGHT	ACCELERATI	YEAR	VARIABLE
Mazda Glc	Custom	31	4	91	68	1970	17.6	82	MODEL
Mazda Glc	Delux	32.8	4	78	52	1985	19.4	78	MODEL
Mazda Glc	4	34.1	4	91	68	1985	16	81	MODEL
Mazda Glc	Custom L	37	4	91	68	2025	18.2	82	MODEL
Mazda Glc		46.6	4	86	65	2110	17.9	80	MODEL

Table 3.15: Part of KWIC table of 'Renault'.

KEYWORD		MPG	CYLINDERS	DISPLACEME	POWER	WEIGHT	ACCELERATI	YEAR	VARIABLE
Renault	12 (Sw)	26	4	96	69	2189	18	72	MODEL
Renault	12Tl	27	4	101	83	2202	15.3	76	MODEL
Renault	18I	34.5	4	100	0	2320	15.8	81	MODEL
Renault	5 Gtl	36	4	79	58	1825	18.6	77	MODEL
Renault	Lecar Delux	40.9	4	85	0	1835	17.3	80	MODEL

Table 3.16: Part of KWIC table of 'Volkswagen'.

KEYWORD		MPG	CYLINDERS	DISPLACEME	POWER	WEIGHT	ACCELERATI	YEAR	VARIABLE
Volkswagen	411 (Sw)	22	4	121	76	2511	18	72	MODEL
Volkswagen	Type 3	23	4	97	54	2254	23.5	72	MODEL
Volkswagen	Dasher	25	4	90	71	2223	16.5	75	MODEL
Volkswagen	1131 Delux Sedan	26	4	97	46	1835	20.5	70	MODEL
Volkswagen	Dasher	26	4	79	67	1963	15.5	74	MODEL
Volkswagen	Super Beetl	26	4	97	46	1950	21	73	MODEL
Volkswagen	Model 111	27	4	97	60	1834	19	71	MODEL
Volkswagen	Rabbit Custom	29	4	97	78	1940	14.5	77	MODEL
Volkswagen	Rabbit	29	4	90	70	1937	14	75	MODEL
Volkswagen	Dasher	30.5	4	97	78	2190	14.1	77	MODEL
Volkswagen	Rabbit	29.5	4	97	71	1825	12.2	76	MODEL
Volkswagen	Scirocco	31.5	4	89	71	1990	14.9	78	MODEL
Volkswagen	Jetta	33	4	105	74	2190	14.2	81	MODEL
Volkswagen	Rabbit L	36	4	105	74	1980	15.3	82	MODEL
Volkswagen	Rabbit Custom Diesel	43.1	4	90	48	1985	21.5	78	MODEL

Table 3.17: Part of KWIC table of 'Vw'.

KEYWORD		MPG	CYLINDERS	DISPLACEME	POWER	WEIGHT	ACCELERATI	YEAR	VARIABLE
Vw Rabbit		29	4	90	70	1937	14.2	76	MODEL
Vw Rabbit	Custom	31.9	4	89	71	1925	14	79	MODEL
Vw Rabbit		41.5	4	98	76	2144	14.7	80	MODEL
Vw Rabbit	C (Diesel)	44.3	4	90	48	2085	21.7	80	MODEL

Table 3.18: Part of KWIC table of 'Honda Civic'.

KEYWORD		MPG	CYLINDERS	DISPLACEME	POWER	WEIGHT	ACCELERATI	YEAR	VARIABLE
Honda Civic		24	4	120	97	2489	15	74	MODEL
Honda Civic	(Auto)	32	4	91	67	1965	15.7	82	MODEL
Honda Civic		33	4	91	53	1795	17.4	76	MODEL
Honda Civic	Cvcc	33	4	91	53	1795	17.5	75	MODEL
Honda Civic	1300	35.1	4	81	60	1760	16.1	81	MODEL
Honda Civic	Cvcc	36.1	4	91	60	1800	16.4	78	MODEL
Honda Civic		38	4	91	67	1965	15	82	MODEL
Honda Civic	1500 Gl	44.6	4	91	67	1850	13.8	80	MODEL

A comparison between two selected groups of cars, where the first group (group (A)) has high *MPG* values and the other group (group (B)) has low *MPG* values, is done. Considering the following charts that are based on the above-mentioned steps we can, afterwards, make some conclusions about fuel efficient cars:

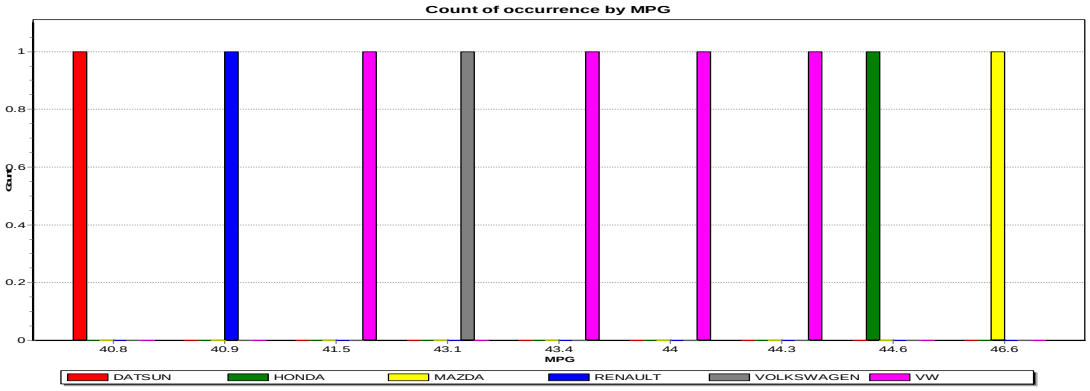

Figure 3.19: Count of occurrence by MPG (*group* (*A*)).

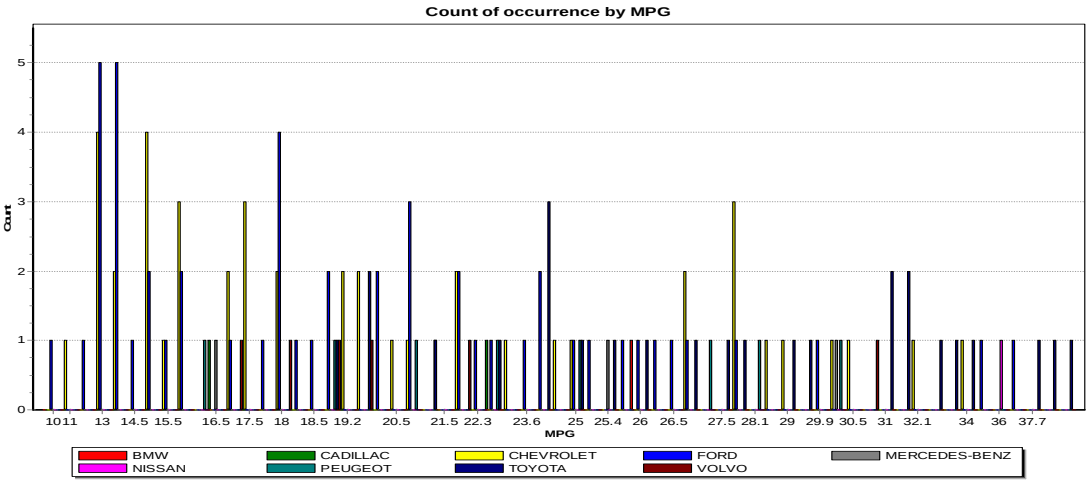

Figure 3.20: Count of occurrence by MPG (*group* (*B*)).

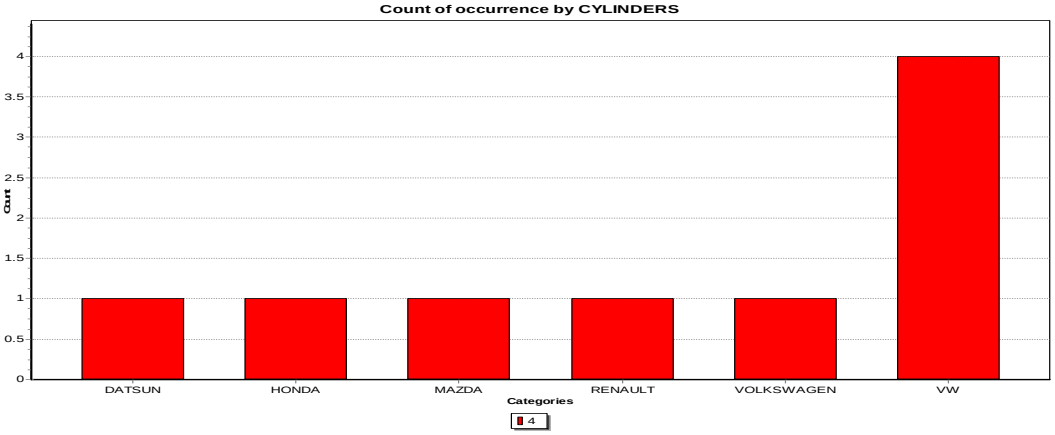

Figure 3.21: Count of occurrence by cylinders (*group* (*A*)).

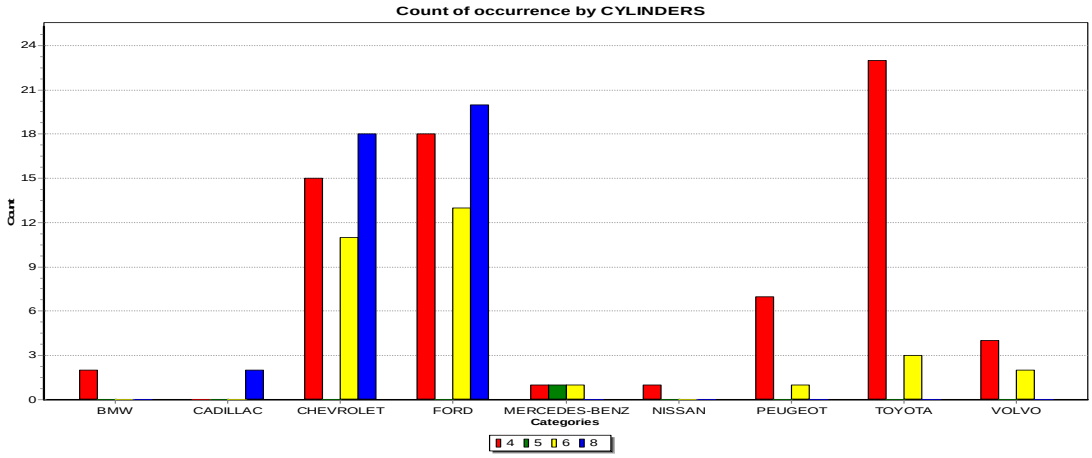

Figure 3.22: Count of occurrence by cylinders (*group* (*B*)).

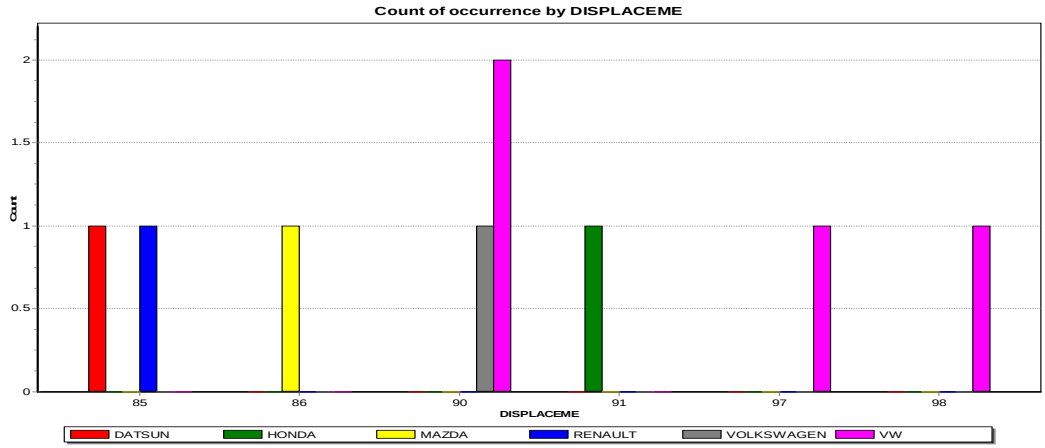

Figure 3.23: Count of occurrence by displacement (*group* (*A*)).

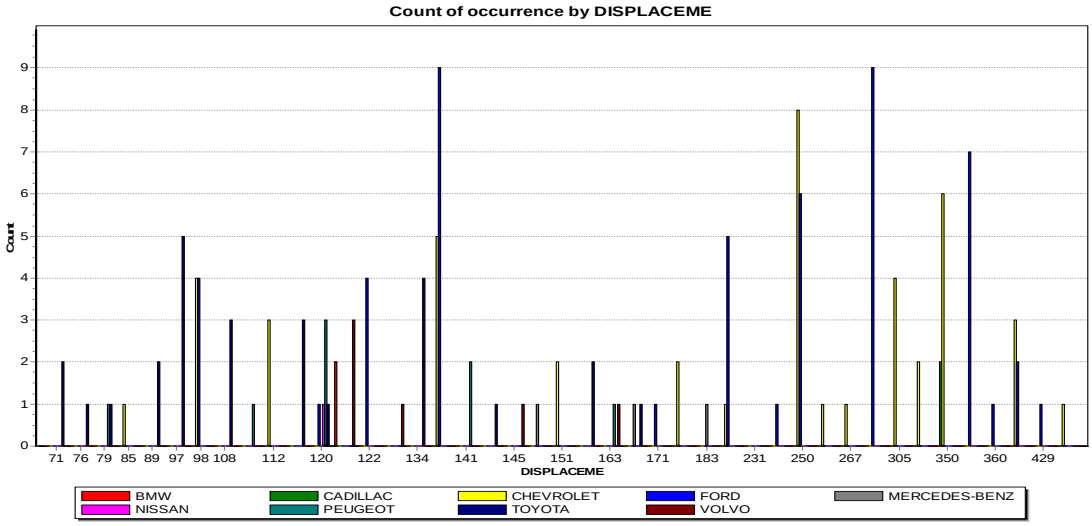

Figure 3.24: Count of occurrence by displacement (*group* (*B*)).

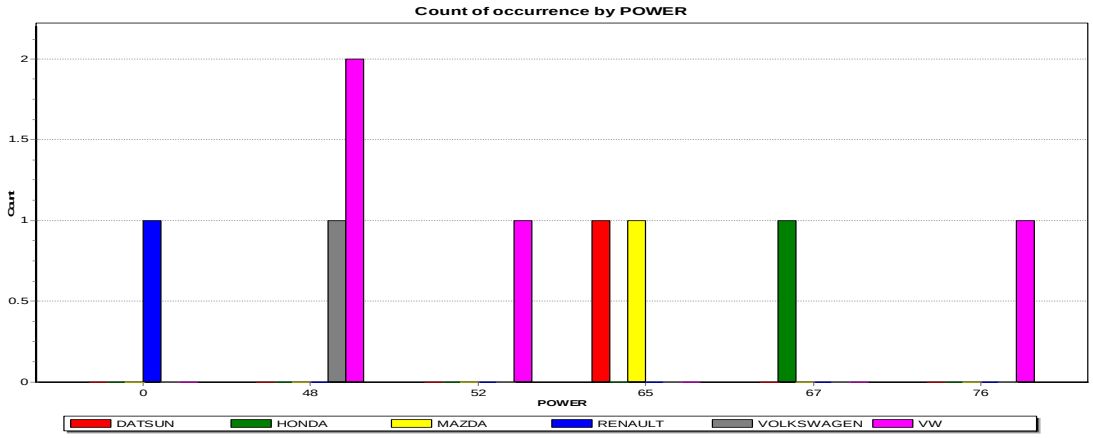

Figure 3.25: Count of occurrence by power (*group* (*A*)).

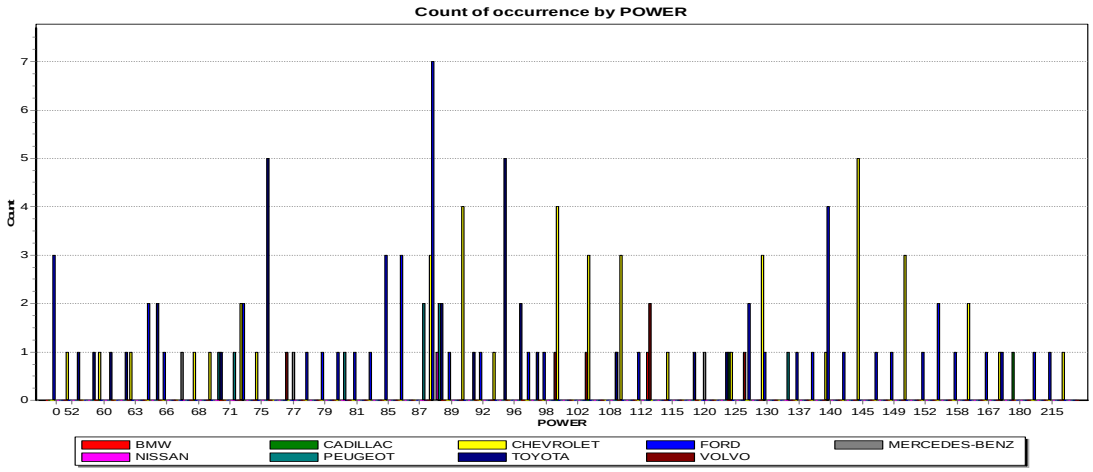

Figure 3.26: Count of occurrence by power (*group* (*B*)).

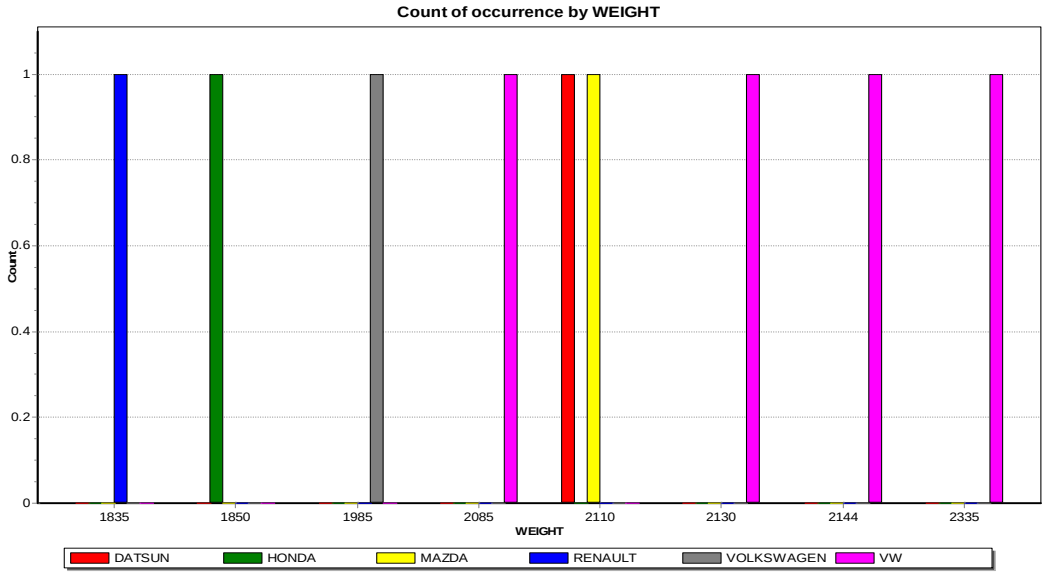

Figure 3.27: Count of occurrence by weight (*group* (*A*)).

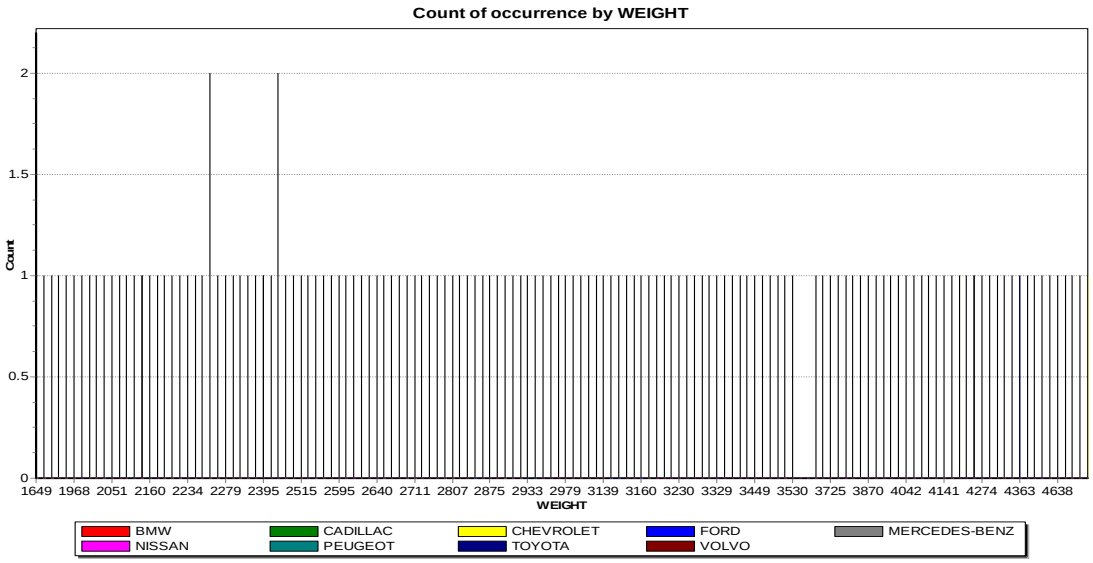

Figure 3.28: Count of occurrence by weight (group (B)).

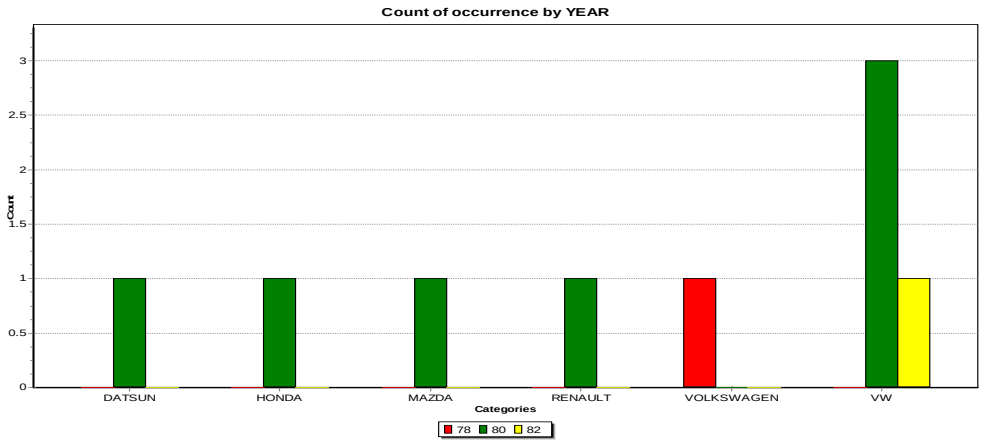

Figure 3.29: Count of occurrence by year (group (A)).

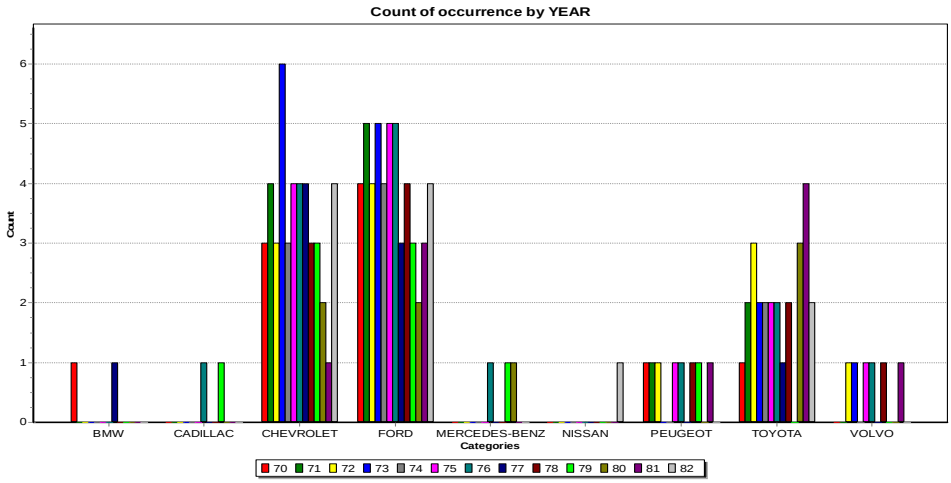

Figure 3.30: Count of occurrence by year (group (B)).

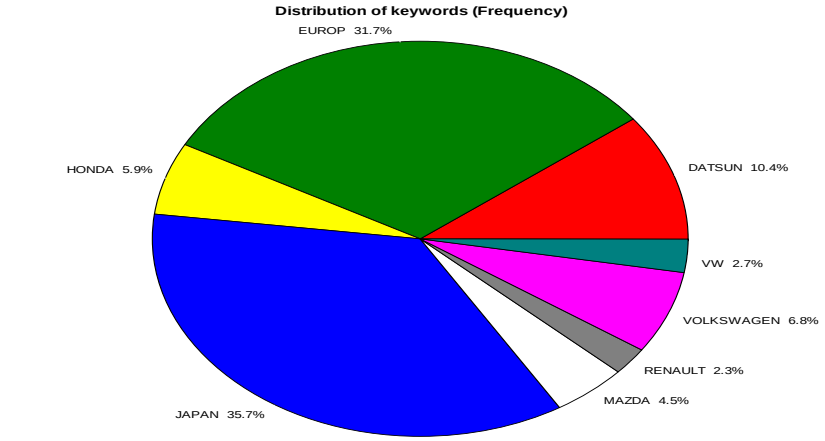

Figure 3.31: Distribution of keywords (*Frequency*) (*group* (*A*)).

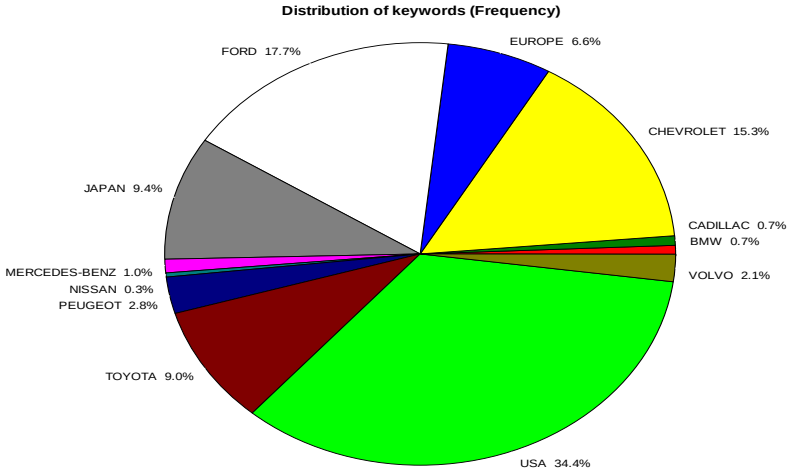

Figure 3.32: Distribution of keywords (*Frequency*) (*group* (*B*)).

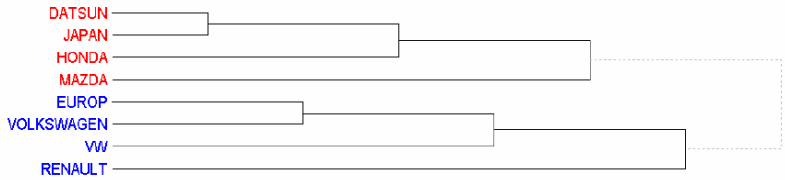

Figure 3.33: Agglomeration order Jaccard's coefficient (*occurrence*) (*group* (*A*)).

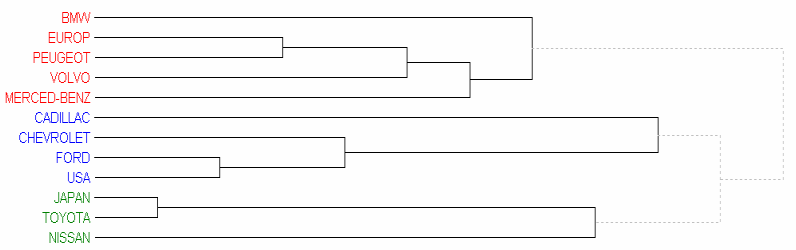

Figure 3.34: Agglomeration order Jaccard's coefficient (*occurrence*) (*group* (*B*)).

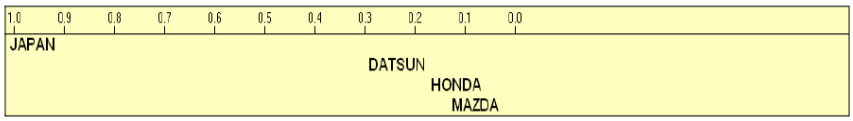

Figure 3.35.a: Proximity plot for Japan (*group* (*A*)).

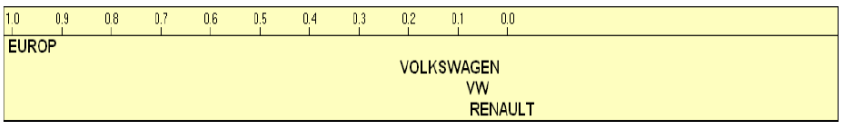

Figure 3.35.b: Proximity plot for Europe (*group* (*A*)).

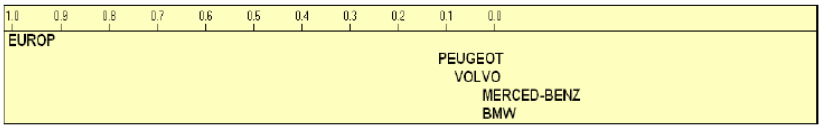

Figure 3.36.a: Proximity plot for Europe (*group* (*B*)).

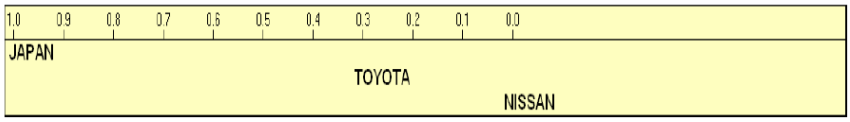

Figure 3.36.b: Proximity plot for Japan (*group* (*B*)).

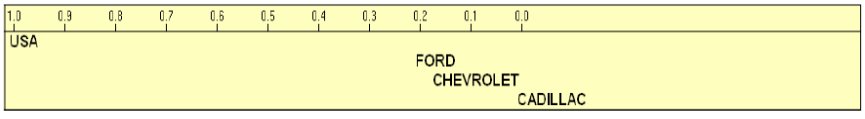

Figure 3.36.c: Proximity plot for USA (*group* (*B*)).

3.2.6 .1 Conclusions

The following conclusion about fuel efficient cars compared with non fuel efficient ones can be obtained:

- they all have only four cylinders
- they have about twice smaller average cylinder displacement and horse power
- they are lighter and younger on average
- they all were manufactured in Japan and Europe.
- non fuel efficient cars accelerate much slower (they take longer to reach 100 mph). Fuel consumption efficiency improves as manufacturers improve engine technology, which increases acceleration.

CHAPTER 4

CONCLUSION

The explosive growth of stored information in almost every area of human activity has created a great demand for new, powerful tools for turning data into useful knowledge.This problem of information overload is further aggravated due to the unstructured, textual data form of the majority of the data. Text expresses a vast, rich range of information, but encodes this information in a form that is difficult to decipher automatically. Perhaps for this reason, there has been little work in text data mining to date, and most people who have talked about it have either conflated it with information retrieval or have not made use of text directly to discover heretofore unknown information. Researchers often spend substantial time creating tools necessary to access, clean and model textual data. Many tools exist but are often specific to the application for which they were designed or are not publicly available.

Text mining is a burgeoning new technology for discovering knowledge from text data. As with any emerging research area, there is still no established vocabulary for *text mining*, a fact which can lead to confusion when attempting to compare results and techniques. As the most natural form of storing information is *text*, text mining is believed to have a commercial potential higher than that of data mining. In fact, a recent study indicated that 80% of a company's information is contained in text documents. However, identifying all key facts and issues, opinion leaders, and ground-breaking ideas, then putting that information in the hands of the right knowledge workers in a company will lead to the best decisions, and, in time, to a successful organization overall.

Quality is the most important determinant of profit, and quality improvement increases consumer demand for products and services.

In this thesis, the capabilities of text mining as a burgeoning quality improvement tool in the manufacturing industry and service sector were studied.

Manufacturing is an application area where text mining can provide significant competitive advantage. Quickly solving product yield and quality problems in a complex manufacturing process is becoming increasingly more difficult.
Text mining technology can increase product yield and quality by quickly finding and solving these problems.
Services account for more than 75% of the *GDP* in most developed and developing countries. It is therefore important to improve quality in this vital sector of the economy. Therefrom the significance of using any tool, such as text mining, that aids this target.

4.1 Future Work

In this thesis, a text mining operation, text categorization, was presented. Essentially, popular machine learning methods borrowed by data mining, organized, here, in terms of text mining, and inherent concepts in text categorization highlighted. Similar work can be done on additional machine learning methods and text categorization schemes. Other text mining operations may be discussed.
Improving the mathematical methods and techniques involved in text mining would lead to optimize the accuracy of the results obtained and, consequently, to maintain a competitive edge in every phase of the customer life cycle.
Regarding the experimental part, testing effects of text preprocessing approaches, document (text) representation schemes, and dimension reduction techniques on the accuracy of the results obtained and choosing the best of ones would lead to better consequences.

4.2 Thesis Summary

The amount of textual information, both in documents and Web pages, is huge and increasing. With the so-called information overloading problem, caused by the growing availability and heavy use of electronic textual information, there has been increasing interest in tools that can help in organizing and describing the large

amount of online textual information for later retrieval and use. The Text Mining field was born to address the huge demand for mining large amounts of text automatically. Text mining techniques can range from simple one (e.g., arithmetic averages) to those with intermediate complexity (e.g., linear regression, clustering and decision trees) and highly complicated ones such as neural network. Dealing with applied analysis, especially approximation theory, and applied probability, as well as linear algebra, the mathematics of computation, and other areas of analysis, its techniques are intimately linked with applied mathematics. Given its broad applicability, text mining has seen widespread application in many industries ranging from finance, bioinformatics, pharmaceuticals, telecommunications and others.

The focus of this thesis study was about the capabilities of text mining as a burgeoning quality improvement tool in the manufacturing industry and service sector.

Quality is the most important determinant of profit, and quality improvement increases consumer demand for products and services. On the other hand, quality improvement is one of the typical ways a firm could benefit from text mining. To maintain or improve the quality of a firm products and/or services, it has to start by enhancing the quality of its information. Text mining lets it collect data from unstructured sources such as warranty notes, customer contact centers and online information and merge it with data it already has, creating a more powerful knowledge base for better decision making.

Two experimental case studies regarding the utilization of a text mining tools were carried out that demonstrated approaches in how to deal with text mining in order to enhance the quality of the work in the fields studied.

In the case studies, the applications of various mathematical methods in text mining area mentioned throughout the thesis were accomplished. This comprised *NB*, *k-NN*, *Tfidf*, *CHI*, agglomerative hierarchical clustering using Jaccard's similarity method, leave-one-out, categorization precision, categorization recall, categorization accuracy, micro- and macro-averaging, etc..

REFERENCES

[1] Alpaydin, E., *Introduction to Machine Learning*, The MIT Press, 2004.

[2] American Society for Quality, *Glossary*, http://www.asq.org/glossary/q.html, 2007.

[3] Awad, M., and Khan, L., *Applications and Limitations of Support Vector Machines*, University of Texas at Dallas, USA, 2004.

[4] Bakus, J., and Kamel, M.-S., *Higher Order Feature Selection for Text Classification*, Springer-Verlag London Ltd., 2005.

[5] Baoli, L., Shiwen, Y., and Qin, L., *An Improved k-Nearest Neighbor Algorithm for Text Categorization*, Shenyang, China, 2003, 117-120.

[6] Baoli, L., Yuzhong, Chen, and Shiwen, Y., *A Comparative Study on Automatic Categorization Methods for Chinese Search Engine*, Proceedings of the Eighth Joint International Computer Conference, Hangzhou: Zhejiang University Press, 2002, 117-120.

[7] Basu, A., Watters, C., and Shepherd, M., *Support Vector Machines for Text Categorization*, Dalhousie University, Halifax, Nova Scotia, Canada, Proceedings of the 36th Hawaii International Conference on System Sciences, 2002.

[8] Beall, A.-L., Editor, *SAS Business Report*, 2006.

[9] Berkhin, P., *Survey of Clustering Data Mining Techniques*, Accrue Software, Inc., 2002.

[10] Bolasco, S., Canzonetti, A., Capo, F.-M., della Ratta-Rinaldi, F., and Singh, B.-K., *Understanding Text Mining: A Pragmatic Approach*, Roma, Italy, 2005.

[11] Borgatti, S.-P. *Multidimensional Scaling*, http://www.analytictech.com/networks/mds.htm, 2002.

[12] Braha, D., *Data Mining for Design and Manufacturing*, Springer, 2002.

[13] Brinker, K., *Active Learning with Kernel Machines*, University of Paderborn, Ph.D. Thesis, Paderborn, 2004.

[14] Burges, C.-J.-C., *A Tutorial on Support Vector Machines for Patern Recognition*, 1998.

[15] Callan, J., *Text Data Mining: Automatic Classification*, Carnegie Mellon University, 2004.

[16] Callan, J., *Text Data Mining: Introduction to Text Data Mining*, Carnegie Mellon University, 2005.

[17] Chakrabarti, S., *Mining the Web: Analysis of Hypertext and Semi Structured Data, Morgan Kaufmann,* San Francisco, CA, USA, 2002, 96-99.

[18] Chen, B., *Machine Learning*, 2006.

[19] Chen, P.-H., Lin, C.-J., and Schölkopf, B., *A Tutorial on v-Support Vector Machines*, 2003.

[20] Chiwara, M., Al-Ayyoub, M., Hossain, M.-S., Gupta, R., and Wasilewska, A., *Text Mining*, 2006.

[21] Cho, S.-H., and Eppinger, S.-D., Product Development Process *Modeling Using Advanced Simulation,* 2001.

[22] Choi, B., and Yao, Z., *Web Page Classification: Foundations and Advances in Data Mining*, 2005.

[23] Clarabridge, *Unlocking the Potential of Unstructured Data*, Clarabridge, Inc., 2007.

[24] Cnn Türk, *An Interview with the Kemer Hotels Chain Administrator*, http://www.cnnturk.com.tr/, 2007.

[25] Cristianini, N., and, Shawe-Taylor, J., *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 2000.

[26] Cristianini, N., *Kernel Methods for General Pattern Analysis*, University of California, Davis, USA, 2004.

[27] Cunningham, P., *Introduction to Scheduling Problems- Stochastic Search*, Dublin, 2007.

[28] Cunningham, P., *Tutorial on Support Vector Machines*, Trinity College, Dublin, 2006.

[29] Dahan, E., and Hauser, J.-R., *Managing a Dispersed Product Development Process*, 2000.

[30] Das, G., and Gunopulos, D., *Time Series Similarity and Indexing*, 2003.

[31] Debole, F., and Sebastiani, F., *Supervised Term Weighting for Automated*

Text Categorization, Melbourne, Florida USA, 2003.

[32] Dick, U., *Semi-supervised Learning for Linked Data*, Albert–Ludwigs–Universität Freiburg, 2006.

[33] Dörre, J., Gerstl, P., and Seiffert, R., *Text Mining: Finding Nuggets in Mountains of Textual Data*, Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining (KDD-99), CA, USA, 1999.

[34] Drewes, B., *Some Industrial Applications of Text Mining*, SAS Institute, Heidelberg, 2004.

[35] Dumais, S., Platt, J., and Heckerman, D., *Inductive Learning Algorithms and Representations for Text Categorization*, 1998.

[36] Duverge, B., Ehlers, M., Lawrence, K., and Deptowicz, D., *Warranty Management: Transforming Aftermarket Processes to Improve Product Quality, Reduce Costs and Enhance Customer Loyalty*, USA, 2005.

[37] Edelman and Nielsen BuzzMetrics, *Talking from the inside out: The Rise of Employee Bloggers*, BuzzMetrics, Inc., 2006.

[38] Eirinaki, M., *Web Mining: A Roadmap*, Athens University of Economics and Business, http://:www.db-net.aueb.gr/index.php/corporate/content/download/341/1381/file/NEMIS.pdf, 2007.

[39] El Wakil, M.-M., *Introducing Text Mining*, Cairo University, 2002.

[40] Ender, P., *Multivariate Analysis: Hierarchical Cluster Analysis,* http://www.gseis.ucla.edu/courses/ed231a1/notes2/cluster.html, 1998.

[41] Erdoğan, R.-T., *An Interview with the Prime Minister of Turkey*, Samanyolu TV, 2007.

[42] Eyheramendy, S., Madigan, D., *A Novel Feature Selection Score for Text Categorization*, Proceedings of the Workshop on Feature Selection for Data Mining: Interfacing Machine Learning and Statistics in conjunction with the 2005 SIAM International Conference on Data Mining, Newport Beach, CA, USA, 2005.

[43] Fan, W., Wallace, L., Rich, S., and Zhang, Z., *Tapping into the Power of Text Mining*, 2005.

[44] Feldman, R., Fresko, M., Hirsh, H., Aumann, Y., Liphstat, O., Schler, Y., and Rajman, M., *Knowledge Management: A Text Mining Approach*, 1998.

[45] Feldman, R., *Mining Text Data*, ClearForest Corp. & Bar-Ilan University, Israel, 2003.

[46] Feldman, S., Martin, J.-N., and Modjeska, N.-N., *Redefining Business Seach: A Picture Is Worth a Thousand Documents,* Adopted from Text Mining for Gold in Unstructured Information, 2006.

[47] Fodor, I.-K., *A survey of dimension reduction techniques*, Lawrence Livermore National Laboratory, University of California, 2002.

[48] Forman, G., *An Extensive Empirical Study of Feature Selection Metrics for Text Classification*, 2002.

[49] Forman, G., Editors: Guyon, I., and Elisseeff, A., *An Extensive Empirical Study of Feature Selection Metrics for Text Classification*: Journal of Machine Learning Research, Palo Alto, CA, USA, 3 (2003), 1289-1305.

[50] Fung, G., *A Comprehensive Overview of Basic Clustering Algorithms*, 2001.

[51] Gallaher, M., Link, A., and Petrusa, J., *Measuring Service-Sector Research and Developmen*t, 2005.

[52] Garcke, J., *Support Vector Machine*, 2005.

[53] Gardner, M., Bieker, J., *Data Mining Solves Tough Semiconductor Manufacturing Problems*, Boston, MA, USA, 2000.

[54] Geremew, M., *Machine Learning in Text Categorization*, 2005.

[55] Graaf, R.D., *Assessing Product Development, Visualizing Process and Technology Performance with Race*, PHD, Technische Universiteit Eindhoven, 1996.

[56] Gunn, S.-R., *Support Vector Machines for Classification and Regression*, University of Southampton, 1998.

[57] Gutierrez-Osuna, R., *Introduction to Pattern Analysis*, Texas A&M, University, 2002.

[58] Guyon, I., and Elisseeff, A., *An Introduction to Feature Extraction*, 2006.

[59] Han, E., and Karypis, G., *Centroid-Based Document Classification: Analysis & Experimental Results*, Proceedings of the Fourth European Conference on the Principles of Data Mining and Knowledge Discovery, Lyon, France, 2000, 424–431.

[60] Han, J., and Kamber, M., *Data Mining: Concepts and Techniques*, Morgan Kaufmann, San Francisco, 2000.

[61] Han, J., and Kamber, M., *Data Mining: Concepts and Techniques*, Simon Fraser University, 2000, 15-17.

[62] Han, J., and Kamber, M., *Data Mining: Concepts and Techniques*, University of Illinois, Urbana-Champaign, 2006.

[63] Han, J., *Data Mining: Concepts and Techniques*, University of Illinois at Urbana-Champaign, 2006.

[64] Hand, D., Mannila, H., and Smyth, P., *Principles of Data Mining,* The MIT Press, 2001, 49-172.

[65] Harding, J.-A., M. Shahbaz, Srinivas S., and Kusiak, A., *Data Mining in Manufacturing: A Review,* Journal of Manufacturing Science and Engineering Vol. 128 / 969, 2006.

[66] He, J., Tan, A.-H., Tan, C.-L., *A Comparative Study on Chinese Text Categorization Methods*, 2005.

[67] Hearst, M.-A., *Untangling Text Data Mining*, University of California, Berkeley, CA, USA, 1999.

[68] Hearst, M.-A., *What is Text Mining?*, http://www.sims.berkeley.edu/~hearst/text-mining.html, 2003.

[69] Hotho, A., Nürnberger, A., and Paaß, G., *A Brief Survey of Text Mining*, 2005.

[70] Hulth, A., and Megyesi, B.-B., *A Study on Automatically Extracted Keywords in Text Categorization*, Uppsala University, Sweden, 2006.

[71] Inniss, T.-R., Light, M., Thomas, G., Lee, J.-R., Grassi, M.-A., and Williams, A.-B., *Towards Applying Text Mining and Natural Language Processing for Biomedical Ontology Acquisition*, Arlington, Virginia, USA, 2006.

[72] Jaakkola, T.-S., *Machine learning: lecture 7*, 2004.

[73] Jankowski, N., and Grąbczewski, K., Learning Machines, 2006.

[74] Jin, Z., *Support Vector Machines*, 2006.

[75] Joachims, T. *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*, Proceedings of the European Conference on Machine Learning, Dortmund, Germany, 1998.

[76] Joachims, T., *A Statistical Learning Model of Text Classification for Support Vector Machines*, Sankt Augustin, Germany, 2001.

[77] Kaban, A., *Support Vector Machines Kernel Machines*, The University of

Birmingham, 2005.

[78] Kanaan, G., *KNN Arabic Text Categorization Using IG Feature Selection*, Amman AL-Ahliyya University, Jordan, 2006.

[79] Kao, A., and Poteet, S., *Text Mining and Natural Language Processing: Introduction for the Special Issue*, 2004.

[80] Karanikas, H., and Theodoulidis, B., *Knowledge Discovery in Text and Text Mining Software*, Manchester, UK, 2002.

[81] Kecman, V.., *Support Vector Machines :An Introduction,* The University of Auckland, New Zealand, 2005.

[82] Ko, Y., Seo, J., *Text Categorization for Multi-label Documents and many Categories*, 2005.

[83] Koller, D., and Sahami, M., *Toward Optimal Feature Selection*, Stanford University, CA, USA, 1996.

[84] Kontostathis, A., and Pottenger, W.-M., *Detecting Patterns in the LSI Term-Term Matrix*, Lehigh University, 2002.

[85] Kou, H., *Text Mining*, Georges Gardarin Prism Laboratory, 2000.

[86] Kwek, S., *Classification and Prediction*, 2005.

[87] Lagus, K., *Text Mining with the WEBSOM*, Helsinki University of Technology, Finland, 2000.

[88] Landauer & Littman, *A statistical method for language-independent representation of the topical content of text segments*, University of Colorado, 2005.

[89] Lang, J., *Feature Selection: Do we need it?*, 2005.

[90] Larose, D.-T., *Discovering Knowledge in Data: An Introduction to Data Mining*, Published by John Wiley & Sons, Inc., 2005, 2-147

[91] Lau, K.-N., Lee, K-H., and Ho, Y., *Text Mining for the Hotel Industry*, Cornell Hotel and Restaurant Administration Quarterly, Cornell University, 2005.

[92] Lee, K.-H., *Text Categorization with a Small Number of Labeled Training Examples*, Ph.D. Thesis, the University of Sydney, 2003.

[93] Lei, H., and Govindaraju, V., *Speeding Up Multi-class SVM Evaluation by PCA and Feature Selection*, State University of New York at Buffalo, Amherst, NY, USA, 2005.

[94] Leopold, E., Kindermann, J., *Text Categorization with Support Vector Machines: How to Represent Texts in Input Space?*, Kluwer Academic Publishers, Manufactured in The Netherlands, *Sankt Augustin, Germany,* 2002.

[95] Lewis, D.-D., Yang, Y., Rose, T.-G., Editor: Dietterich, T.-G., *RCV1: A New Benchmark Collection for Text Categorization Research*, Journal of Machine Learning Research 5 (2004) 361-397.

[96] Liu, B., Dai, Y., Li, X., Lee, W.-S., and Yu, P.-S., *Building Text Classifiers Using Positive and Unlabeled Examples*, USA, 2003.

[97] Loh, H.-T., Koh, W.-L., Menon, R., and Leong, C.-K., *A Study of Service Center Records Using Data Mining*, 2002.

[98] Loh, H.-T., Menon, R., Leong, C.-K., Mining of Text in the Product Development Process, 2002.

[99] London: Office for National Statistics, *The UK Service Sector*, Crown copyright, December 2000.

[100] Luz, S., Dimensionality reduction, Trinity College, 2006.

[101] Matsatsinis, N.-F., Ioannidou, E., and Grigoroudis, E., *Customer Satisfaction Using Data Mining Techinques*, 1999.

[102] Megaputer Intelligence, *PolyAnalyst*, http://www.megasysdev.com/webdown/prodlist;jsessionid=a5FhoxjRCk4b, 2006.

[103] Menon, R., *Mining of Textual Databases within the Product Development Process*, Ph.D. Thesis, National University of Singapore, 2004.

[104] Metadata Extraction Project Sponsored by DTIC, *Evaluation of Different Algorithms for Metadata Extraction*, Old Dominion University, 2004.

[105] Michie, D., Spiegelhalter, D.J., Taylor, C.C., *Machine Learning, Neural and Statistical Classification*, 1994.

[106] Mitra, S., and Acharya, T., *Data Mining: Multimedia, Soft Computing, and Bioinformatics*, Published by John Wiley & Sons, Inc., 2003, 1-236.

[107] Moore, A.-W., *Support Vector Machines*, Carnegie Mellon University, 2003.

[108] Music Semiconductors, *Quality and Reliability Handbook*, 6th edition, 2003.

[109] Musicant, D.-R. , and Mangasarian, O.-L., *Nonlinear Data Discrimination via Generalized Support Vector Machines*, University of Wisconsin, Madison,

2006.

[110] Nahm, U.-Y., *Text Mining with Information Extraction*, Ph.D. Thesis, The University of Texas at Austin, 2004.

[111] Nakov, P., *SIMS 290-2: Applied Natural Language Processing*, 2004.

[112] Neshatian, K., and Hejazi, M.-R., *Text Categorization and Classification in Terms of Multi-Attribute Concepts for Enriching Existing Ontologies*, 2004.

[113] Nitta, H., *Invigorating Japan's Service Sector*, 2004.

[114] Novovičová, J., Malík, A., Pudil, P., *Feature Selection using Improved Mutual Information for Text Classification*, 2003.

[115] Novovičová, J., Malík, A., *Information-Theoretic Feature Selection Algorithms for Text Classification*, Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, 2005.

[116] OECD, *OECD Employment Outlook,* 2001.

[117] Ograjenšek, I., *Applying Statistical Tools to Improve Quality in the Service Sector*, 2002.

[118] Özgür, A., Özgür, L., Güngör, T., *Text Categorization with Class-Based and Corpus-Based Keyword Selection*, Boğaziçi University, Turkey, Springer-Verlag Berlin Heidelberg 2005.

[119] Parr Rud, O., *Data Mining Cookbook: Modeling Data for Marketing, Risk, and Customer Relationship Management*, Published by John Wiley & Sons, Inc., New York, 2001, 1.

[120] Pedersen, R.-U., *Using Support Vector Machines for Distributed Machine Learning*, Ph.D. Thesis ,University of Copenhagen, Copenhagen, Denmark, 2004.

[121] Pekar, V., Krkoska, M., Staab, S., *Feature Weighting for Co-occurrence-based Classification of Words*, 2002.

[122] Péladeau, N., and Stovall, C., *Application of Provalis Research Corp.'s Statistical Content Analysis Text Mining to Airline Safety Reports: A Technology Demonstration of the SimStat/WordStat Software by Provalis Research Corporation at JetBlue Airways,* 2005.

[123] Pilászy, I., *Text Categorization and Support Vector Machines*, Budapest University of Technology and Economics, 2005.

[124] Poggio, T., and Smale, S., *The Mathematics of Learning: Dealing with Data*,

2003.

[125] Politecnico di Milano, Dipartimento di Elettronica e Informazione, *A Tutorial on Clustering Algorithms*: *Hierarchical Clustering Algorithms*, http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/hierarchical.html, 2007.

[126] The Principal Financial Group, *Financial services*, http://www.goliath.ecnext.com/coms2/gi_0199-4589101/Text-mining-for-the-hotel.html.

[127] Provalis Research, *Word Stat Content Analysis Module for SIMSTAT & QDA Miner*, Montreal, QC, Canada, 2005.

[128] Rajman, M., and Vesely, M., *From Text to Knowledge: Document Processing and Visualization: A Text mining Approach*, Swiss Federal Institute of technology, Lausanne, Switzerland, 2003.

[129] Rosario, B., *Applied Natural Language Processing*, 2004.

[130] Sahlgren, M., and Swanberg, D., Vector Based Semantic Analysis: Modeling Linguistic Knowledge in Computer Systems, M.Sc. Thesis, University of Stockholm, 2000.

[131] Salman, T., *Support Vectors Machines*, 2005.

[132] Sandu-Popa, I., Zeitouni, K., Gardarin, G., Metais, E., and Nakache, D., *Text Categorization for Multi-label Documents and many Categories*, France, 2005.

[133] SAS Institute Inc., *Clustering Methods*, Cary, NC, USA, http://v8doc.sas.com/sashtml/stat/chap23/sect12.htm, 1999.

[134] Sassano, M., *Virtual Examples for Text Classification with Support Vector Machines*, Fujitsu Laboratories Ltd., Kawasaki, Japan, 2003.

[135] Schölkopf, B., and Smola, A.-J., Learning with Kernels, http://www.learning-with-kernels.org/sections/, 2007.

[136] Schölkopf, B., *Statistical Learning and Kernel Methods*, 2000.

[137] Schönhofen, P., and Benczúr, A.-A., Feature selection based on word–sentence relation, Hungarian Academy of Sciences Lagymanyosi, Budapest, Hungary, 2004.

[138] Sebastiani,F., *Machine Learning in Automated Text Categorization*, Consiglio Nazionale delle Ricerche, Italy, 2002.

[139] Sheikh, L.-M., *Knowledge Discovery and Data Mining: Applications,*

Techniques, and Performance Issues, 2005.

[140] Shin, K., Abraham, A., and Han, S.-Y., *Improving kNN Text Categorization by Removing Outliers from Training Set*, Springer-Verlag Berlin Heidelberg, Chung-Ang University, Seoul, Korea, 2006.

[141] Sify Ltd, *Service sector will power Indian economy*, http://sify.com/news/thethursdayinterview/fullstory.php?id=13678650, 2005.

[142] Singh, N., Hu, C., and Roehl, W.-S., *Text mining; a Decade of Progress in Hospitality Human Resource Management Research: Identifying Emerging Thematic Development*, Philadelphia,, USA, 2007,

[143] Soucy, P., Mineau, G.-W., *Feature Selection Strategies for Text Categorization*, Université Laval, Québec, Canada, Springer-Verlag Berlin Heidelberg 2003.

[144] Spinakis, A., and Chatzimakri, A., *Comparative Study of Text Mining Tools*, Athens, Greece, 2005.

[145] Spinakis, A., Peristera, P., *Text Mining Tools: Evaluation Methods and Criteria*, Athens, Greece, 2003.

[146] Spinakis, A., *Text Mining: A Powerful Tool for Knowledge Management*, 2001.

[147] SPSS Survey Research Solutions for Government and Academic Institutions, *Collect High-Quality Research Data,* 2004.

[148] Sprinkhuizen-Kuyper, I., *Support Vector Machines*, 2006.

[149] Stacks, D.-W., *Dictionary of Public Relations Measurement and Research*, University of Miami, 2007.

[150] Strathmeyer, J., *Automatic Text Categorization*, 2004.

[151] Sun, J.-T., Chen, Zheng, Zeng, H.-J., Lu, Y.-C., Shi, C.-Y., Ma, and W.-Y., *Supervised Latent Semantic Indexing for Document Categorization*, Proceedings of the Fourth IEEE International Conference on Data Mining, 2004.

[152] Suykens, J.-A.-K., Van Gestel, T., De Brabanter, J., De Moor, B., Vadewalle, J., *Least Squares, Support Vector Machines*, World Scientific Publishing Co. Pte. Ltd., Singapore, 2005.

[153] Tan, A.-H., *Text Mining: The State of the Art and the Challenges*, Singapore, 1999.

[154] Tang, B., Shepherd, M., Milios, E., and Heywood, M.-I., *Comparing and Combining Dimension Reduction Techniques for Efficient Text Clustering*, Dalhousie University, Halifax, Canada, 2005.

[155] Thatcher, M.-E., and Oliver, J.-R., *The Impact of Information Technology on Quality Improvement, Productivity, and Profits: An Analytical Model of a Monopolist*, Proceedings of the 34th Hawaii International Conference on System Sciences, 2001.

[156] Tong, L.-H., Menon, R., and Sathiyakeerthi, S., *Analyzing Textual Databases Using Data Mining to Enable Fast Product Development Processes*, 2004.

[157] Tong, L.-H., Menon, R., Sathiyakeerth, S., and Brombacher, A., *Automated Text Classification for Fast Feedback: Investigating the Effects of Document Representation*, Springer-Verlag Berlin Heidelberg 2003.

[158] Two Crows Corporation, *Introduction to Data Mining and Knowledge Discovery,* Third Edition, Two Crows Corporation, Potomac, (U.S.A.), 1999, 1-18.

[159] Udoh, E., and Rhoades, J., *Mining Documents in a Small Enterprise Using WordStat*, Indiana University – Purdue University, Fort Wayne, USA, 2006.

[160] Van Deun, K., and Delbeke, L., *Multidimensional Scaling*, University of Leuven, Belgium, 2002.

[161] Vapnik, V. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[162] Volkovich, Z., Barzily, Z., and Morozensky, L., *On Statistical Models of Cluster Stability*, 2006.

[163] Weber, S.-P., *World Wide Web Document Search Techniques*, M.Sc. Thesis, The University of Texas at Austin, 1999.

[164] Wikipedia, *Quality*, http://en.wikipedia.org/wiki/Quality, 2007.

[165] Wikipedia, *Service*, http://en.wikipedia.org/wiki/Service, 2007.

[166] Wikipedia, *Tertiary Sector of Industry*, http://en.wikipedia.org/wiki/Service_sector, 2007.

[167] Witten, I.-H., *Text Mining*, University of Waikato, Hamilton, New Zealand, 2004.

[168] XLMiner, *Hierarchical Clustering*, http://www.resample.com/xlminer/help/HClst/HClst_intro.htm, 2007.

[169] Yan, J., Liu, N., Zhang, B., Yan, S., Chen, Z., Cheng, Q., Fan, W., and Ma, W.-Y., *Optimal Orthogonal Centroid Feature Selection for Text Categorization*, 2005.

[170] Yang, Y., and Liu, X., *A Re-examination of Text Categorization Methods*, Proceedings of 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1999, 42-49.

[171] Yang, Y., and Pederson, J.-O., *A Comparative Study on Feature Selection in Text Categorization*, 1997.

[172] Yu, E.-S., and Liddy, E.-D., *Feature Selection in Text Categorization Using the Baldwin Effect*, 1999.

[173] Yu, L., Liu, H., *Efficient Feature Selection via Analysis of Relevance and Redundancy*, Arizona State University, AZ, USA, Journal of Machine Learning Research, 5(2004), 1205–1224.

[174] Yuan, L., *Analysing Reliability Problems in Concurrent Fast Product Development Processes*, Ph.D. Thesis, National University of Singapore, 2002.

[175] Zaïane, O.-R., *Principles of Knowledge Discovery in Databases,* University of Alberta, 1999.

[176] Zheng, Z., Wu, X., and Srihari, R., *Feature Selection for Text Categorization on Imbalanced Data*, 2003.