COMPARISON OF DOCUMENT CLASSIFICATION APPROACHES FOR TURKISH TEXTS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Computer Engineering

by
Özlem Ece ÇOBANOĞLU

July 2015
İZMİR

We approve the thesis of **Özlem Ece ÇOBANOĞLU**

Examining Committee Members:

Instr. Dr. Burak Galip ASLAN
Department of Computer Engineering, İzmir Institute of Technology

Assist. Prof. Dr. Gürcan ARAL
Department of Physics, İzmir Institute of Technology

Assist. Prof. Dr. Korhan KARABULUT
Department of Software Engineering, Yaşar University

13.07.2015

Instr. Dr. Burak Galip ASLAN
Supervisor, Department of Computer Engineering
İzmir Institute of Technology

Prof. Dr. Halis PÜSKÜLCÜ
Head of the Department of Computer Engineering

Prof. Dr. Bilge KARAÇALI
Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGEMENTS

First and foremost, I am sincerely grateful to my supervisor, Dr. Burak Galip ASLAN, who has supported me throughout my thesis with his patience. His guidance helped me in all the time of research and writing of this thesis.

I would like to thank Assist. Prof. Dr. Korhan KARABULUT for supporting my research and helping me to develop my background in Natural Language Processing.

I am heartily thankful to my father, mother and brother for all their endless love and encouragement. They were always supporting me with their best wishes.

Lastly, and most importantly, I would like to thank my husband, İsmail YÜREK. Many thanks to him for his continuous encouragement and efforts. This thesis would not have been completed without him.

ABSTRACT

COMPARISON OF DOCUMENT CLASSIFICATION APPROACHES FOR TURKISH TEXTS

Internet usage is exponentially growing day by day. This rapid growth in Internet usage leads to an explosion in the number of electronic documents being produced daily. The huge bulk of documents make it difficult accessing the necessary and relevant information. Due to lack of logical organization, retrieval and processing of the desired information from huge number of documents becomes a complex and time consuming task with human effort. Therefore, document classification is significant task to manage and process the documents.

In this thesis, the performance of different classification approaches produced from several algorithms is thoroughly evaluated. The main goal of the thesis is to determine the best combination of document preprocessing steps and classification algorithms. Different feature weighting, construction and selection methods are experimented on Turkish documents.

Stemmed and original words and their bi-gram and tri-gram forms are used to construct the features which represent the documents. The effects of several weighting algorithms and the combination of feature selection and weighting algorithms on 3 different classification approaches are interpreted. The performance of 216 different classification process combinations are analyzed.

Experimental results show that C4.5 (C4.5 Decision Tree) classification algorithm has the highest accuracy results in 95% of the results. SVM (Support Vector Machine) algorithm produces the closest results to C4.5 and it provides the highest accuracy in 5% of the experimental results. NB (Naive Bayes) algorithm has always the lowest accuracy rate in these 3 different classification algorithm results.

ÖZET

TÜRKÇE METİNLER İÇİN DOKÜMAN SINIFLANDIRMA YAKLAŞIMLARININ KARŞILAŞTIRILMASI

Gün geçtikçe yaygınlaşan internet kullanımıyla beraber elektronik belgelerde hızlı bir artış yaşanmaktadır. Belgelerin çoğu herhangi bir mantıksal yapıda olmadığı için insan gücü ile bu belge yığınlarının içinden istenilen bilgiye ulaşmak karmaşık ve zaman alıcı bir iştir; bu nedenle belgeleri hızlı bir şekilde düzenlemek, yönetmek ve işlemek için belge sınıflandırma önemli bir işlemdir.

Bu tezde, Türkçe belgelerde farklı algoritmaların kullanılması ile birden fazla sınıflandırma yaklaşımının performansları değerlendirilmektedir. Tezin başlıca hedefi belge önişleme adımları ve sınıflandırma algoritmaları arasındaki en iyi kombinasyonun belirlenmesidir.

Belgeleri temsil eden özelliklerin oluşturulmasında belgede geçen kelimelerin doğrudan kendileri, kökleri, bi-gram ve tri-gram formları kullanılmıştır. Bu özellik setlerine farklı ağırlıklandırma, seçim ve sınıflandırma algoritmalarının uygulanmasıyla 216 deneysel sonuç elde edilmiştir.

Elde edilen deneysel sonuçlara göre, C4.5 (C4.5 Decision Tree) sınıflandırma algoritması sonuçların %95'inde en yüksek doğruluk değerine sahiptir. SVM (Support Vector Machine) algoritması C4.5'e en yakın sonuçları üretmektedir; ve bu sonuçların %5'inde en yüksek doğruluk değerini vermektedir. NB (Naive Bayes) algoritması ise bu 3 farklı sınıflandırma algoritması içinde her zaman en düşük doğruluk oranına sahip olduğu gözlemlenmiştir.

I would like to dedicate my thesis to my family,
Hüseyin Çobanoğlu,
Gülay Çobanoğlu,
Erdem Çobanoğlu,
İsmail Yürek

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

LIST OF ABBREVIATIONS

BPNN	Back–Propagation Neural Network
BW	Boolean Weighting
C4.5	C4.5 Decision Tree
CBFS	Correlation Based Feature Selection
CVM	Concept Vector Model
FN	False Negatives
FP	False Positives
ID3	Iterative Dichotomiser 3
IDF	Inverse Document Frequency
IG	Information Gain
IR	Information Retrieval
K-NN	K-Nearest Neighbors
LPP	Locality Pursuit Projection
LSA	Latent Semantic Analysis
LSI	Latent Semantic Indexing
LS-SVM	Least Square Support Vector Machine
NB	Naive Bayes
NLP	Natural Language Processing
NTCIR	NII Test Collection for IR System
RF	Random Forest
SLSI	Supervised LSI
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
TN	True Negatives
TP	True Positives
TREC	Text Retrieval Conference
VSM	Vector Space Model
WEKA	Waikato Environment for Knowledge Analysis
X^2 statistic	Chi Square Statistic

CHAPTER 1

INTRODUCTION

Usage of Internet is continuously increasing considerably day by day. This rapid increase in Internet usage caused an expansion in data accumulation. As a result of this uncontrollable growth, the World Wide Web (WWW) has become a major resource for preserving and reaching any kind of information.

The data explosion and the rapid expansion of WWW lead to excessive information load. The information is primarily stored in text documents. Retrieving necessary and relevant information from these huge mass of documents is a very complicated and time consuming task. Useful information easily turns into troublesome information, because there is no logical organization and hierarchy in documents. People have to cope with very large amount of documents every day and they have to spend too much time to find and process the required information among the documents, so building up a logical structure on documents with human effort is almost impossible. Besides, extracting relevant knowledge from this document heap is a challenging task. As a solution to this challenge, it is vital to develop methods so as to automate processing and managing huge amounts of documents. Information Retrieval (IR) is a suitable methodology for automated organization of information, so document classification has a crucial role in this area.

Automatic document classification is one of the most significant ways of finding, managing, filtering and processing information. Fundamentally, document classification is the process of assigning predefined classes to documents depending on their content. Documents can be electronic publications, electronic books, email messages, news, digital libraries, academic articles, Web pages, and so on. Several machine learning algorithms are proposed to enhance automatic document classification.

Rich textual information is stored in documents, therefore text understanding, processing and analyzing is essential in order to extract the valuable information. Natural Language Processing (NLP) is a very challenging and popular subject area that is being researched and studied continuously. However, there are not enough studies about NLP for Turkish language. NLP research in Turkish is seriously difficult because of the basic differences of agglutinative languages such as extreme usage of affixes.

NLP is a branch of computer science, artificial intelligence, and linguistics that deals with analyzing, understanding and generating the languages that humans use naturally. Automatic text summarization, machine translation, natural language generation, question answering, sentiment analysis, word sense disambiguation, document classification are some of the important tasks in NLP.

NLP suggests strong techniques for automatically classifying documents. NLP assumes that the documents in different classes discriminate themselves by features of the natural language such as word structure, word frequency, word stem and natural language structure in each document.

In this thesis, we worked on a document classification system. The goal of the thesis is to evaluate the performance of different classification approaches generated from several algorithms and strategies. After preparing the document collection, several preprocessing methods are applied. Then, different document representation models are used to present texts and different feature weighting algorithms are calculated. After weighting the features in the document, feature selection strategies are considered. Finally, several machine learning algorithms are run. Many combinations of these techniques are carried out and the outputs of each combination is measured based on evaluation metrics. To sum up, the main idea is to determine the best combination of document preprocessing steps, document representation model, feature weighting, feature selection methods and machine learning algorithms.

This thesis is organized as follows: In Chapter 2, the background information and the literature review about the document classification are explained. In Chapter 3, the implementation details, the methods and strategies applied to classify the documents in the thesis are described. Chapter 4 is the chapter in which the experimental results and evaluation of these results are discussed. Finally, Chapter 5 concludes this study.

CHAPTER 2

BACKGROUND

2.1. Related Work

In literature, there are several researches about document classification. Considerable amount of studies in document classification domain are applied to English texts. Li and Park (2007) modeled the text with LSI (Latent Semantic Indexing) and traditional vector space model (VSM). Then, the results of classifiers for each model were compared. The results indicated that LSI is faster than the VSM and also the classification results were better than VSM based on performance evaluation measures.

Wang et al. (2003) implemented an automatic web document classification system named WebDoc. Zhang et al. (2008) suggested a web classification system depending on a least square support vector machine (LS-SVM) with latent semantic analysis (LSA). The researchers compared the performance of LS-SVM, SVM and k-NN (k-Nearest Neighbors). Mohamed (2007) designed an automatic document classification system. He combined several parameters and design decisions to see their effects on automatic classifiers. He applied these different cases to neural networks.

Li and Jain (1998) performed 4 separate classification methods. NB, k-NN, decision tree and subspace method. Li and Jain (1998) worked on 3 classifier combination methods: simple voting, dynamic classifier selection and adaptive classifier combination. The experimental studies showed that the combination of classifiers did not always result in higher classification accuracy values.

Deng and Peng (2006) applied SVM classifier based on concept features of documents. Their experimental results indicated that concept vector model (CVM) carries out better than the traditional term based VSM.

Wang et al. (2008) offered a new document classification approach by using locality pursuit projection (LPP) and SVM. They applied different combinations of classifiers such as LPP and SVM, LSI and SVM and LSI and k-NN. Then, they evaluated and compared the results based on performance measures.

Lee et al. (2006) studied on the performance of the supervised and unsupervised techniques for multilingual text classification. SVM was selected as a supervised method and LSI was selected as an unsupervised method. Then, they compared and evaluated the results based on performance measures.

Ishii et al. (2006) came up with a new approach for the document classification. They applied LSA and k-NN algorithms. The results showed that the new approach achieves higher accuracy in the classification.

Sun et al. (2004) suggested Supervised LSI (SLSI) method for document classification. Again, Li and Park (2007) suggested a new text classification approach by using LSA and a back–propagation neural network (BPNN).

Shi et al. (2008) imposed LSI in order to model web pages and then implemented classification algorithm. The classification algorithm was the k-NN combined with SVM.

Liang (2004) was interested in multi-classification problem using SVM. He compared the complexity, construction and performance of different SVM multi-classifiers. Lee et al. (2006) applied LSI technique to classify multi-language texts.

Magatti et al. (2009) implemented a system for topic extraction and automatic document classification. The software system found the main topics in the document collection. When users confirmed the topic extracted from the document, this topic was used to assist the automatic document classifier. These topics were used as labels for each new document.

The most discussed problem in the classification of documents is how to represent the text. Amasyali et al. (2012) applied LSI and different text representation methods to classify documents. Taghva and Vergara (2008) indicated that the use of font and capitalization as features enhances precision and recall. They focused on different feature selection approaches. Schenker et al. (2003) applied a different approach for text representation. Instead of using traditional vector-based model, a graph-based model was used for document representation. They compared the 2 model based on the classification accuracy. They stated that graph-based k-NN algorithm performs better than the vector-based k-NN model.

Preprocessing, feature extraction, dimension reduction and text representation models are main and very important steps of text classification. There are different researches that focus on these main steps of classification.

Cataltepe et al. (2007) showed how different stemming methods affect the performance of document classification. They used the shortest and the longest root of the word, discarded silent letters of the word and also took the first 3 or 4 letters of the word as a stem in order to observe the results. Also, Tufekci et al. (2012) studied on the effects of stemming in Turkish document classification. They considered the effects of selecting different lengths and types of the words as features. They observed that noun type and the maximum length of word stems as features are more successful than others in document classification. They applied different classifier algorithms and the experimental results indicated that NB performs better than SVM, C4.5 and RF (Random Forest) classification methods.

Toraman et al. (2011) concentrated on the effect of preprocessing steps in document classification. Several combinations of these preprocessing steps were evaluated in this paper. Each combination was called as a setup. The researchers focused on discovering the best setup for document classification. They applied these setups both in Turkish and English and also on different domains such as e-mails and news so as to observe impact of the preprocessing steps in different cases. Again, Torunoglu et al. (2011) analyzed how the preprocessing methods affect the classification of Turkish texts. They studied deeply on preprocessing methods like stop word eliminating, stemming and word weighting for Turkish text classification on different datasets.

Feature extraction is another important side of the document classification. Yildiz et al. (2007) presented a new approach for extracting features in text classification. They represented text with lower dimensional space. Each class is symbolized as a dimension, so the number of classes were used as the dimensions of the vector. Instead of words weights in texts, weights in class are used in this approach. Then, class weights of the words are collected in the document and normalized to create new feature vectors. The paper indicates that NB outperforms all other applied classification methods.

There are lots of different text presentation methods in literature. Amasyali et al. (2010) worked on different text representation methods. Amasyali et al. (2010) prepared an elaborative paper that represents text with various text representation methods. Texts are generally presented with bag of words model. In this model, each dimension is equivalent to a word or n-gram. Amasyali and Beken (2009) introduced the first research that dimensions are located in a semantic space. Words are placed in semantic space based on their meanings. They classified Turkish news into 5 categories. Their experimental results showed that their approach was more successful than the bag of words text representation model. Amasyali et al (2012) prepared a detailed comparison paper about text representation methods for Turkish text classification. 17 different representation methods were applied in text classification and then their success was evaluated. They stated that the paper was the most comprehensive study for Turkish text classification. Their results indicated that n-grams are more successful among the different text representation methods.

In Turkish, there are different studies that examine different classification algorithms. Guran et al. (2009) applied several classifiers on different data sets which were presented with unigram, bi-gram and tri-gram words. Then, they analyzed the effects of n-gram in Turkish text classification methods by surveying n-gram models. Amasyali and Yildirim (2004) built a system that automatically classifies Turkish news. Again, Uzundere et al. (2008) implemented a document classification system. Amasyali and Diri (2006) applied character n-grams and used different classification algorithms for determining the author of the text, genre of the text and gender of the author. This research is the first detailed study on Turkish text classification by using n-grams. Once again, Amasyali et al. (2006) classified documents based on their author. Different feature vectors were created and then several classification algorithms were applied. They reported that NB and SVM are more successful than C4.5 and RF.

2.2. Document Classification

Document classification is the process of assigning documents to predefined classes such as sport, entertainment, health and etc. Classifying the documents automatically is a very important task in many different real-word problems. Huge amount of unclassified documents like academic documents, news, archival records and scientific articles need to be classified to manage the data easily.

The logic behind the document classification is assigning a class to a document. Let $X = (X_1, X_2, X_3,\ldots., X_N)$ represent the set of documents. N refers to the total number of documents in the set X. X_i denotes the i^{th} document to be classified. Predefined classes are $C = (C_1, C_2, C_3,\ldots., C_m)$ where m refers to the total number of predefined classes. The document x_i is assigned to class c_j depending on the function f. f is formulated as, $f: x_i \rightarrow c_j$. The classifier algorithm f defines the class c in the class set of the document x (Ramasundaram & Victor, 2013). In this thesis, SVM, NB and C4.5 classifier algorithms are implemented.

2.2.1. Document Preprocessing and Representation

Text documents are unstructured data. Before applying any machine learning algorithm, documents are transformed into a suitable form for computing. Documents are represented in such a way that a classifier can read and process the document.

A generally used model in NLP to represent a text document is VSM. VSM is the most common document representation way which is suggested by Salton (1968). VSM represents each document as a vector of features. It gets all the words in a text document and puts them in a vector to represent features of the document. Each feature in vector is weighted based on the number of occurrences in the document. The order of the features is ignored but the number of occurrences of each term is important in the model. The occurrence of each feature is used to train a classifier. Each dimension of the text document corresponds to a unique feature. A feature can be a word or any representation of the text.

Document representation is the final task in preprocessing the documents. Also, it has a vital role in preprocessing steps, because if the representation is more relevant to the document, the accuracy of classification results will be higher.

2.2.1.1. Datasets

A dataset is a collection of data. The dataset includes document, query and relevant judgment set. Document set is a collection of documents. Query set is a set of questions asking the IR system for results. Relevant judgment set includes methods to calculate the relevance between result sets and queries.

A dataset has a significant role in IR, because it is used to see how well an IR system performs. Also, the dataset is used to compare the performance of the IR systems, search algorithms and search strategies with the other systems. There are several standard datasets in IR. These standard datasets are Text Retrieval Conference (TREC), Reuters, 20Newsgroups, Gov2, NII Test Collection for IR System (NTCIR).

2.2.1.2. Document Parsing

Documents in the dataset contain raw data. Getting valuable information from this raw data requires parsing processes. Html mark-up tags, punctuation marks, numbers, spaces and symbols are removed from the data so as to get the text.

2.2.1.3. Stop words

Stop words are the word groups which do not have linguistic meaning. For example, in the Turkish language, words such as “acaba”, “ancak”, “belki”, “zaten” etc. are known to be useless words in a text. Stop words are used frequently in a language; they are also disregarded by search engines. The reason why search engines disregard the stop words is these words are present at almost every text; therefore, they do not provide any positive effect on search results.

2.2.1.4. Stemming

Stem is the main smallest meaningful piece of a word. Stemming is the act of analyzing and reducing the word to its root form without considering the context of the word. The aim of stemming is to reduce the inflectional forms and derivationally related forms of a word to a common base. Mostly, morphological variants of words have similar semantic interpretations. But, stemmers work on a single word without knowledge of the context; therefore stemmers can not treat differently to words which have different context.

2.2.1.5. N-Grams

N-gram is one of the most basic techniques in NLP. N consecutive character sequence is called as n-gram. If character sequence size N is one, then n-gram is referred to as a unigram. If N is equal to two, then n-gram is referred to as a bi-gram and if N is equal to three, n-gram is referred to as a tri-gram.

For instance, characters of bi-gram and tri-gram representations of the word “bilgi” are as follows:

Bigram: _b bi il lg gi i_

Trigram: _bi bil ilg lgi gi_

2.2.1.6. Term Weighting

The documents are represented as vectors in VSM. The effectiveness of the VSM depends on term weighting. Term weighting is a vital point in document classification. Terms can be a word, phrase or any representation unit to identify the text. Term weight is related to each term in the text, because each term has different importance. There are several term weighting algorithms so as to discriminate one document from the others. In this section, Boolean Weighting (BW), Term Frequency Weighting (TF) and Term Frequency-Inverse Document Frequency Weighting (TF-IDF) algorithms are described.

2.2.1.6.1. Boolean Weighting

BW is the easiest and basic term weighting technique. In this technique, the algorithm considers the presence or absence of the term in a document. If the term exists in the document, the weight of a term is assigned to be 1, otherwise 0. BW assigns equal importance to every word that appears in a document. Let assume that tf$_i$ is the frequency of term i in a document. Then;

$$w_i = \begin{cases} 1, & if\ tf_i > 0 \\ 0, & if\ tf_i < 0 \end{cases} \quad (2.1)$$

2.2.1.6.2. Term Frequency Weighting

TF weighting depends on the number of occurrences of the term in the document. This algorithm counts how many times the term occurs in a document. The count of occurrences of a term in the document indicates the importance of the term in the document. The weight of a term is equal to the number of times the term exists in the document. The purpose of this weighting algorithm is to make the frequent words more important than the others.

Let assume w_i is the weight of i^{th} term of a document. Then, the weight of a term i in a document is calculated as follows:

$$w_i = tf_i \tag{2.2}$$

2.2.1.6.3. Term Frequency - Inverse Document Frequency Weighting

TF-IDF algorithm is used to find the importance of a term in a document collection. The logic behind this algorithm is based on how often the term exists in multiple documents. BW and TF weighting algorithms do not show the number of occurrences of a term throughout all the documents in the document collection.

TF-IDF is a statistical algorithm which is a combination of term frequency and inverse document frequency. TF measures how frequently a term occurs in a document. Inverse document frequency (IDF) measures how important a term is in a document. TF-IDF weight gets the highest value when term t exists frequently in a small number of documents. This means the term t has high discriminating power in document collection. TF-IDF value gets lower when the term t exists fewer times in many documents. The weighting result gets the lowest value when the term t occurs in all documents.

Let assume N is the total number of documents in the document collection and N_i is the number of documents in the collection where term i occurs. Then, the importance of a term in a document collection is calculated as follows:

$$w_i = tf_i \, . log\left(\frac{N}{N_i}\right) \tag{2.3}$$

2.2.2. Dimensionality Reduction

The number of documents in a dataset collection affects the dimension of the vector space that the machine learning algorithm works on. The number of features in document collection may be more than thousands. Vector space requires dimension proportional to the number of features that exist in the document. Working on high dimensional vector space matrices increases the model complexity for classifiers. Time and computational complexity increase based on the dimension size. Due to the time and computational complexity, before applying any machine learning algorithm to classify text documents, dimension reduction process is performed. Feature selection is one of the effective ways that reduces the dimension of the space.

2.2.3. Feature Selection

Feature selection is a critical problem for document classification. Before any classification work, one of the most significant tasks is feature selection. Owing to the high dimensionality of text features, it is important to select critical features in document classification. The main goal of the feature selection algorithms is to determine the most relevant minimum set of features for the classification process. The final distribution of data for each class is as similar as possible to the actual distribution of the data for each feature. Some terms are more likely to be associative to the class distribution than other terms.

Feature selection reduces computational complexity and saves the time. In text classification, too many features affect the classification performance. Feature selection method eliminates the redundant and irrelevant data and gets a subset of actual features, and then the dataset becomes more efficient for classification. Important features result in higher accuracy in document classification. Important features co-occur with a particular class and generally do not co-occur with other classes. Unimportant features exist across nearly in all classes and they have no discriminative power for that class. Due to the importance of feature selection task, many algorithms have been suggested in literature. In this thesis, IG (Information Gain), X^2 statistic (Chi Square Statistic) and Correlation Based Feature Selection (CBFS) algorithms are applied.

2.2.3.1.1. Information Gain

IG is one of the widely used feature selection algorithms. IG algorithm is implemented when the term existence status is known. IG is a measure of importance of the feature for predicting the presence of the class. Let P_i be the wide scale probability of class i, and $p_i(w)$ be the probability of class i. The document involves the word w. Let F(w) be the fraction of the documents involving the word w. The IG measure I(w) for a given word w is stated as follows (Aggarwal & Zhai, 2012):

$$I(w) = -\sum_{i=1}^{k} P_i . log(P_i) + F(w) \ . \sum_{i=1}^{k} p_i(w). log\big(p_i(w)\big) + \big(1 - \ F(w)\big) \qquad (2.4)$$

$$. \sum_{i=1}^{k} (1 - \ p_i(w)). log\big(1 - p_i(w)\big)$$

The characterizing power of the word w increases while IG value is increasing. If dataset used for classification comprises n documents and x words, the complexity of the IG algorithm is O(n.x.k).

2.2.3.1.2. Chi Square Statistic

The X^2 statistic calculates the value that shows the relationship between a word w and a particular class i. Assume that n is the total number of documents in dataset. $p_i(w)$ is the conditional probability of class i for documents that contains w. P_i is the global fraction of documents including the class i. F(w) is the global fraction of documents that includes the word w. Then, X^2 statistic of the word and class is stated as follows:

$$x_i^2(w) = \frac{n\,.\,F(w)^2.\,(p_i(w) - P_i)^2}{F(w).\left(1 - F(w)\right).P_i.\,(1 - P_i)} \tag{2.5}$$

2.2.3.1.3. Correlation Based Feature Selection

CBFS puts feature subsets in order based on a correlation evaluation as seen from the name of the algorithm. The algorithm assigns high scores and chooses the features which are intensely correlated with the class. The algorithm eliminates the features that have low correlation with the class. Hall & Smith (1998) present this algorithm to evaluate the merit of feature subsets. The algorithm is a heuristic for measuring the worth of a subset of features. CBFS is used to determine the best feature subset in the feature set. Correlation coefficients are used to predict correlation between subset of features and class, and also correlations among the features. CBFS equation is stated as follows:

$$r_{zc} = \frac{k\overline{r_{zi}}}{\sqrt{k + k - (k-1)\overline{r_{ii}}}} \tag{2.6}$$

where r_{zc} is the correlation between the summed feature subsets and the class variable, k is the number of subset features, $\overline{r_{zi}}$ is the average of the correlations between the subset features and the class variable, and $\overline{r_{ii}}$ = is the average inter-correlation between subset features (Hall, 1999).

2.2.4. Classification Algorithms

2.2.4.1. Support Vector Machine

SVM is a powerful computational supervised learning method for solving machine learning problems. SVM was firstly introduced by Vladimir N. Vapnik and

Corinna Cortes in 1993. SVM is a supervised learning algorithm which analyzes data and recognizes patterns used for classification and regression analysis.

SVM is one of the major learning algorithms for document classification. SVM achieves high classification rates and SVM is very effective in high dimensional spaces (Torunoglu et al, 2011). The main idea is to convert the data into a higher dimensional space. Then, method tries to find the optimal hyperplane in the space while ensuring the margin between classes is maximal. To give an example, as shown from Figure 2.1, the first hyperplane H_1 does not separate the classes. The second hyperplane H_2 separates the classes, however with a small margin. But the third hyperplane H_3 separates the classes with the maximum margin.

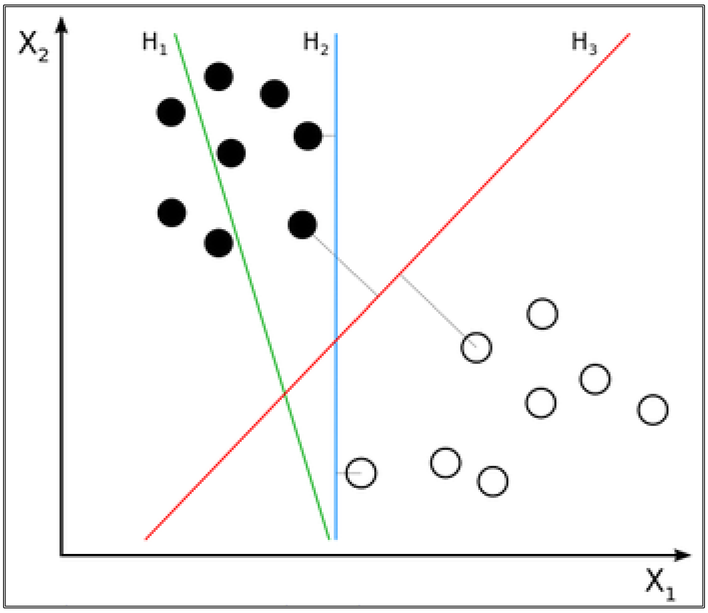


Figure 2.1. The optimal hyperplane H3 in the space
(Source: Wikipedia)

SVM creates a model so as to represent the training examples as points in a dimensional space. These points are $A = \pi r^2$ separated by the hyperplane. The points lying on the hyperplane boundaries are called support vectors. Next, after maintaining labeled training data, SVM uses this model in order to foresee the side of a new point in the space depending on the hyperplane. SVM puts the new point into one class or the other. This approach makes SVM algorithm a binary linear classifier. However, SVM can influentially implement a non-linear classification as well. This non-linear classification is called as kernel trick. Kernel trick implicitly maps the inputs into high dimensional spaces.

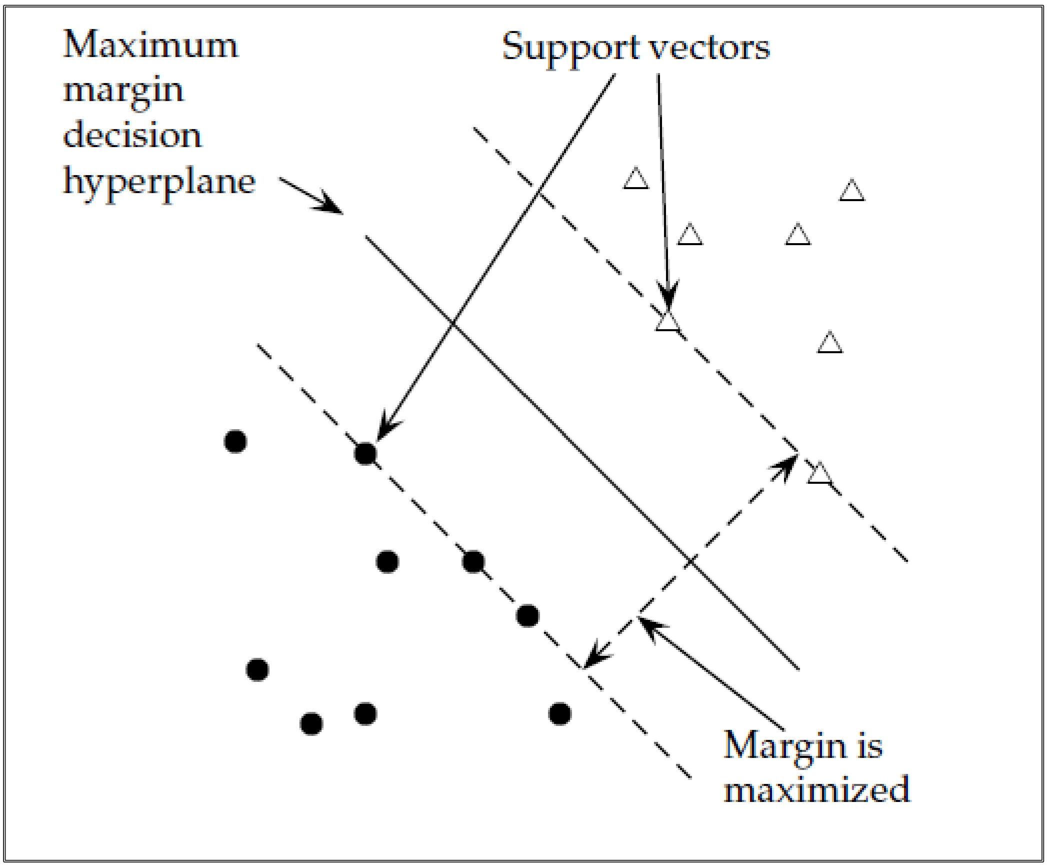


Figure 2.2. The support vectors are the 5 points right up against the margin of the classifier
(Source: Aggarwal & Zhai, 2012)

Finally, SVM is widely used in real life problems. Text, image, protein, cancer classification and hand-written character recognition can be given as example problems which are solved by SVM.

2.2.4.2. Naive Bayes

NB classifier is a probabilistic classifier. The NB classifier applies Bayes' theorem. The classifier is named after Thomas Bayes who proposed Bayes Theorem. The classifier supposes that the occurrence of a feature of a class is unrelated to the occurrence of other words in the document. Also, NB assumes position of a feature in the document does not provide any useful information about class of the document. As a result of these suppositions, the classifier's name is called as "Naive".

NB classifier is used in several different fields such as the diagnosis of diseases (Kazmierska & Malicki, 2008), the classification of RNA sequences in taxonomic studies (Wang et al., 2007) and spam filtering (Sahami et al, 1998). In particular, NB is a significant machine learning algorithm in document classification, because it is easy to implement, robust, fast and accurate. Manning et al. (2008) explained that NB algorithm calculates the probability values so as to assign class labels in document classification. It needs a small amount of training dataset. Furthermore, the training time of NB classifier is significantly small.

2.2.4.3. C4.5 Decision Tree

C4.5 is an algorithm that is used to generate decision trees. C4.5 was firstly introduced by Ross Quinlan (1993). Again, Ross Quinlan invented an algorithm which generates a decision tree from a dataset called Iterative Dichotomiser 3 (ID3). C4.5 algorithm improves ID3 algorithm. C4.5 algorithm avoids overfitting of data. It determines how deeply to grow a decision tree. C4.5 copes with both continuous and discrete attributes, and also with missing attribute values as well. C4.5 successfully selects an appropriate attribute selection measure.

C4.5 is a supervised learning algorithm. A set of training data is given to algorithm as a pair like object and a class. The algorithm analyzes the training data and develops the classification model as a decision tree to classify the test data correctly.

The C4.5 algorithm constructs a decision tree using a set of training data. This decision tree partitions the feature space into two regions. The partitioning process continues until each region includes a single class data, therefore C4.5 is called as a recursive algorithm. C4.5 uses information gain ratio measure so as to choose an instance that partitions the data set into smaller data subsets. Information gain ratio is used as a splitting value. The instance with highest information gain ratio is selected as the splitting instance.

J48 is an implementation of the algorithm C4.5. J48 is an open source Java application. This application is located in WEKA (Waikato Environment for Knowledge

Analysis) data mining tool. This algorithm is easy to implement and the output of the algorithm can be easily understood and interpreted; therefore this algorithm is widely used in machine learning methods and also it is a very popular classification method used in document classification.

CHAPTER 3

METHODOLOGY

3.1. Document Classification Process

Document classification process has many sequential tasks. Before applying machine learning algorithms to classify the documents, preprocessing phases, representation of documents and attribute selection are generally implemented. In the thesis, all the tasks are completed for document classification which is shown in Figure 3.1.

All the tasks in Figure 3.1 can be logically divided into 3 subsections. Reading the document collection and then dataset preparation is the first step in our document classification process. After preparing the dataset, the next step is to eliminate the words which have not any discriminative power to classify the documents. When the documents are ready to be represented, the terms can be generated. Terms in the document are represented in 6 different forms. A stemmer is used to get the root of the word. The first form is the stem of the word. Then, the second form is the original word that means the word is got from the document as how it is seen in the text. Also, the original and the stemmed forms of the words are represented as n-grams. Next, the term generation task is completed in 6 different forms. This task can be called as term preparation task. The 6 different forms are shown in Table 3.1 below**:**

Table 3.1. 6 Different document representation ways

	Term Generation		
	Word	**Bi-gram**	**Tri-gram**
Original Word	Original	Original + Bi-gram	Original + Tri-gram
Stemmed Word	Stem	Stem + Bi-gram	Stem+ Tri-gram

Secondly, after generating the terms, the document is represented with VSM. In order to represent the documents as vectors, term-documents matrices are created. While creating the matrices, 3 different term weighting algorithms are applied. The classifier algorithms are directly applied to these weighted terms. Moreover, 3 different feature selection algorithms are applied to these weighted matrices so as to reduce the dimensions of the vector. The classifier algorithms are applied to the weighted matrices and selected feature matrices with weighted terms. The task that is completed from term generations to classification algorithms implementation can be called as term-document matrix creation.

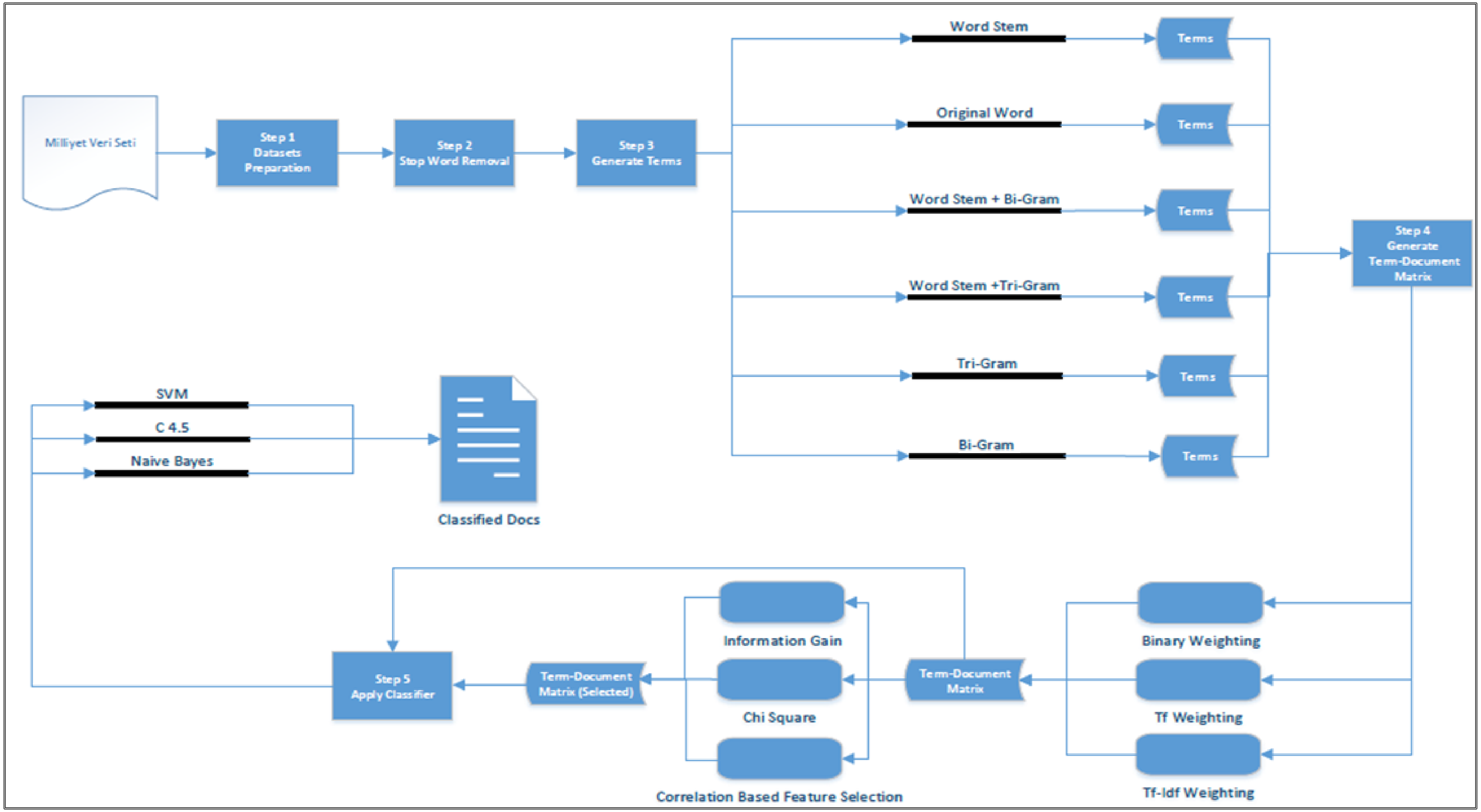


Figure 3.1. Document classification process

Finally, when all the subsections are completed, 3 different machine learning algorithms are applied to classify the documents. The last task in the document classification process can be called as training and testing. After this task is completed, the results are obtained for evaluation and comparison.

To sum up, the document classification process operated in the thesis is clearly seen in Figure 3.1. The process is split into 3 different sections called as term preparation, term-document matrix creation, training and testing. In the following part, all the tasks are explained in detail.

3.2. Term Preparation

Classification result accuracy is directly affected by the implementation of preprocessing phases. Hence, preprocessing phases have significant role in document classification process. In this thesis, until term generating phase, different processes are performed consecutively. Figure 3.2 is called as term preparation section. This figure is an illustration of performed processes. Figure 3.2 indicates that the first step is dataset preparation.

In document classification system, the documents are taken from the Turkish newspaper, because this thesis purely focuses on the Turkish documents.

In Bilkent University, an IR group prepared a dataset called Milliyet Test Collection. Bilkent IR Group intends to implement effective IR tools on the Turkish language. They created a common dataset that contains Turkish news from the newspaper Milliyet. After some official correspondence, we are entitled to download and use this dataset.

In this thesis, Milliyet test collection is used in order to train and test the classification system. The document collection contains 408.305 documents. The size of the document collection is about 1.65 GB. All the documents are Turkish news articles.

The dataset contains 9 different classes: “Güncel”, “Sanat”, “Yaşam”, “Dünya”, “Ekonomi”, “Magazin”, “Siyaset”, “Sağlık” and “Spor”. In the document collection, the class name is generally the last word of the news. All the dataset is read, then every article is put in its related folder named as the related class name. The news in this document collection is formatted using an XML schema. After analyzing the structural elements in documents, the content is obtained. Figure 3.3 shows the number of news for each class after putting them in order.

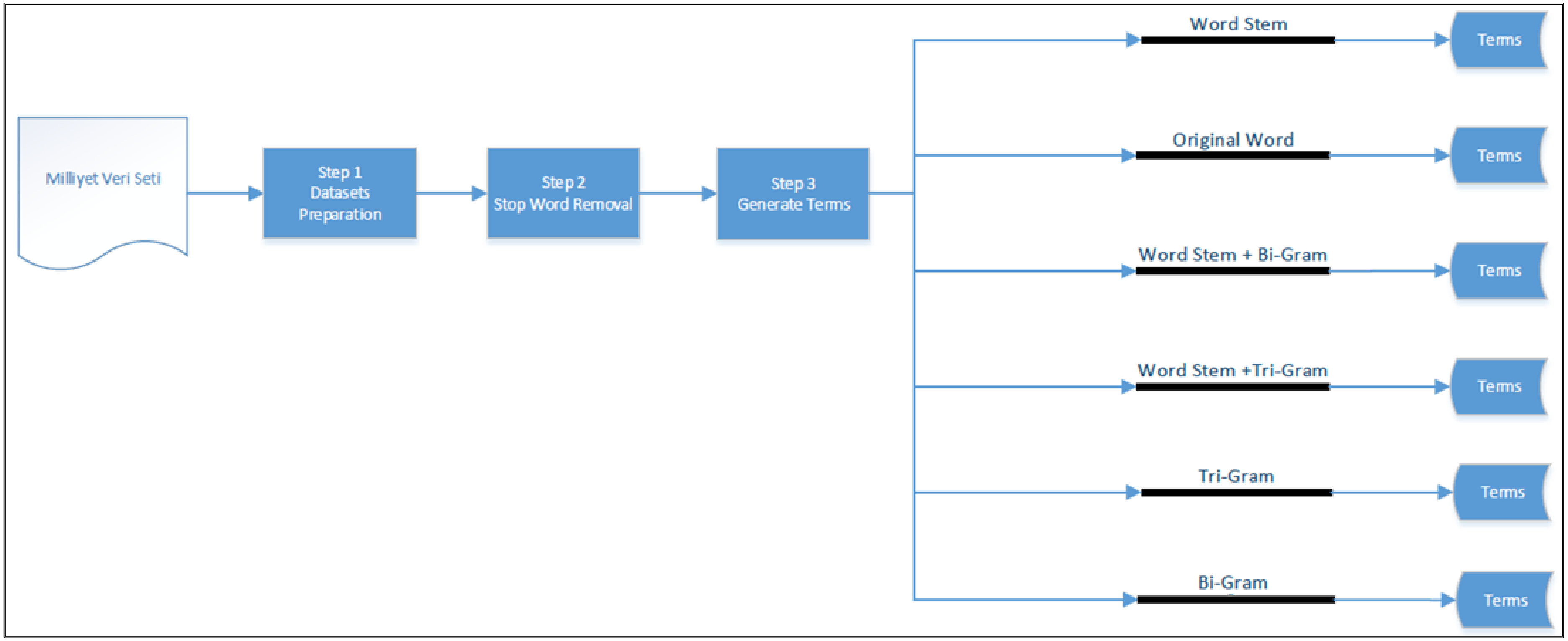


Figure 3.2. Term preparation section

Unfortunately, there is no content in 852 articles of test collection and 349.900 articles in the collection have no class information. Unknown classes and empty files are eliminated from the document collection. As a result, the count of news remained to work on is 57.553. This size of the remained news is nearly 240 MB.

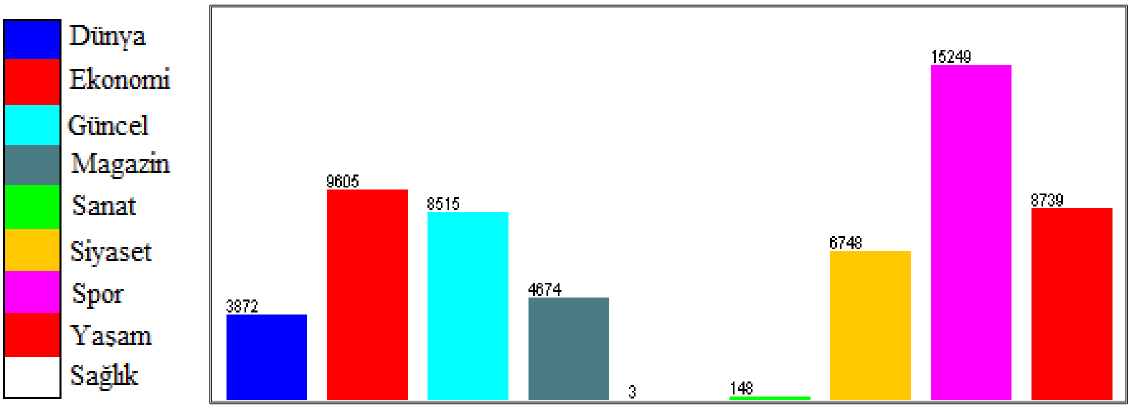
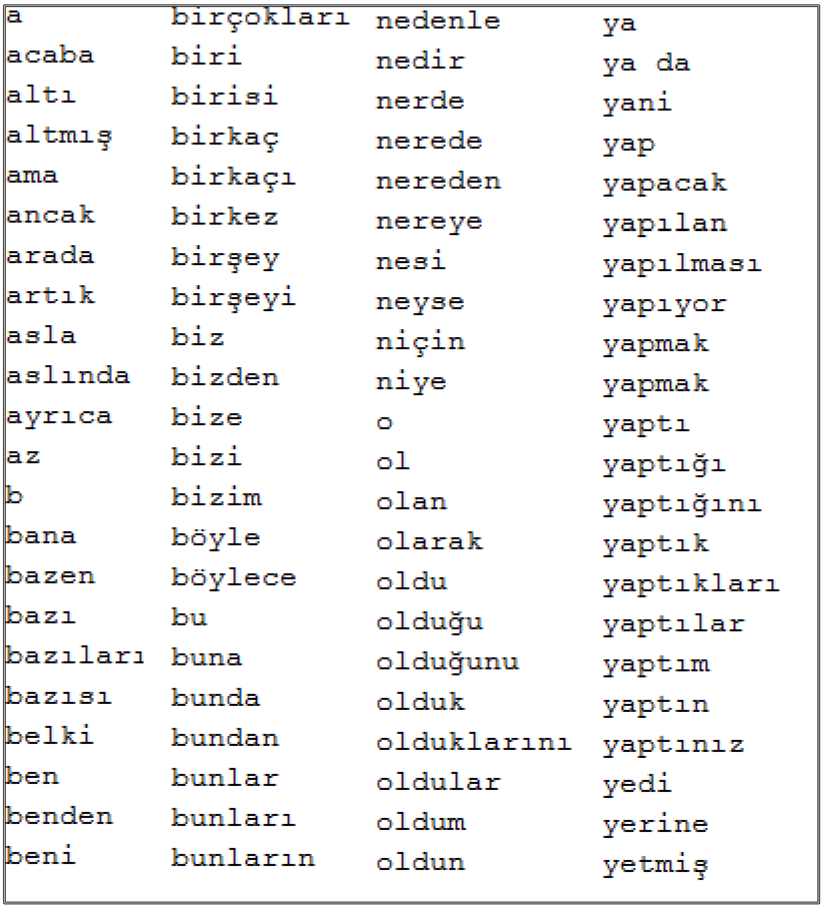


Figure 3.3. The distribution of the news in the classes in document collection

The second step of the data preparation section is stop word elimination. Stop words are encountered very frequently. These words are generally eliminated in IR systems. Because they have poor characterizing power about the class of the text document and they are useless information about the content. Stop word removal reduces the dimensionality of the feature vector. The stop word list used in the thesis during document classification process contains 356 words and is given in Appendix A. Figure 3.4 shows a portion of the stop word list.



a
acaba
altı
altmış
ama
ancak
arada
artık
asla
aslında
ayrıca
az
b
bana
bazen
bazı
bazıları
bazısı
belki
ben
benden
beni
birçokları
biri
birisi
birkaç
birkaçı
birkez
birşey
birşeyi
biz
bizden
bize
bizi
bizim
böyle
böylece
bu
buna
bunda
bundan
bunlar
bunları
bunların
nedenle
nedir
nerde
nerede
nereden
nereye
nesi
neyse
niçin
niye
o
ol
olan
olarak
oldu
olduğu
olduğunu
olduk
olduklarını
oldular
oldum
oldun
ya
ya da
yani
yap
yapacak
yapılan
yapılması
yapıyor
yapmak
yapmak
yaptı
yaptığı
yaptığını
yaptık
yaptıkları
yaptılar
yaptım
yaptın
yaptınız
yedi
yerine
yetmiş

Figure 3.4. Portion of the stop word list used

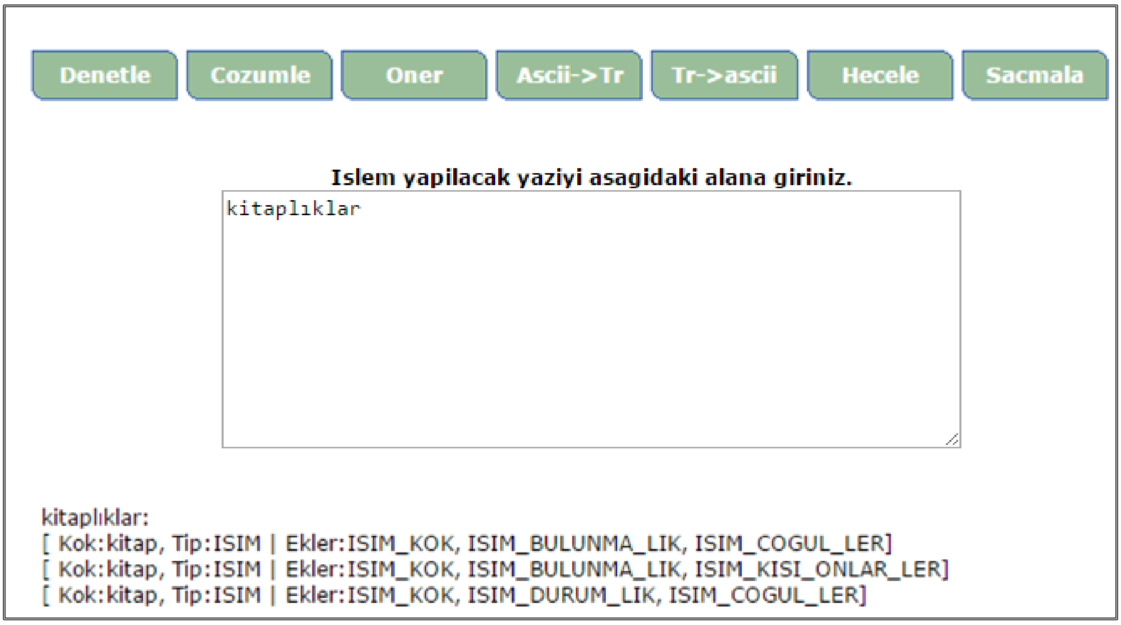
After removing stop words from the document collection, the third step starts. Step 3 is about generating the terms. In this thesis, text documents are represented in 6 different forms. The same dataset, but 6 different models present the dataset collection. This means, each section after generating the terms will be repeated 6 times depending on the text representation models.

Stem of the word is one of the 6 forms. Stem of the word is generated by using NLP methods. NLP operations in Turkish are problematic, due to the lack of open computing libraries. Linguistic processing for stemming is usually done by an additional plug-in component and a few of such components exist, both commercial and open-source. There exists almost no usable open source library for Turkish except Zemberek (Akın and Akın, 2007).

Zemberek is one of the very interesting and important projects oriented around the Turkish language. It is an open-source NLP library and toolset programmed and designed completely in Java programming language for Turkish. This project started in

1999 with a name Tspell and designed as a prototype in C++. In 2004, project is started to be coded in Java and the name of Tspell is changed to Zemberek. Zemberek provides basic NLP operations such as spell checking, morphological parsing, stemming, word construction, word suggestion and converting words written only using ASCII characters. Now, Zemberek library is officially used as spell checker in Open Office Turkish version and Turkish national Linux Distribution Pardus as a stemmer.

However, when Zemberek is tested and analyzed, it is obviously seen that there are some missing parts that affect stemming negatively especially about Turkish idioms and phrases. As a result of being an agglutinative language and being used so many affixes in Turkish; stemmers do not work always truthfully.



Islem yapilacak yaziyi asagidaki alana giriniz.

kitaplıklar

kitaplıklar:
[Kok:kitap, Tip:ISIM | Ekler:ISIM_KOK, ISIM_BULUNMA_LIK, ISIM_COGUL_LER]
[Kok:kitap, Tip:ISIM | Ekler:ISIM_KOK, ISIM_BULUNMA_LIK, ISIM_KISI_ONLAR_LER]
[Kok:kitap, Tip:ISIM | Ekler:ISIM_KOK, ISIM_DURUM_LIK, ISIM_COGUL_LER]

Figure 3.5. Zemberek example for a Turkish word

After getting the stem of the word with Zemberek, all the document collection is represented as the stem of the word. The next representation way is n-grams. Bi-gram and tri-gram representations of the stemmed word are another ways to represent text documents. Moreover, original words in the text document are used. No stemmer is used on the text. Every word in the document is taken with its affixes. For instance, the word “kitaplıklar” is taken as kitaplıklar in the original word form, whereas the stemmed word form of the “kitaplıklar“ is taken as “kitap”. Again, in the original form of the word, bi-gram and tri-gram representations are applied. Finally, 6 different forms of the words in the document are gathered.

Briefly, at the end of the step 3 in the term preparation section, all the terms which will be used are ready to be processed in the next sections. Thus, term-document matrices can be created by using the generated terms.

3.3. Term-Document Matrix Creation

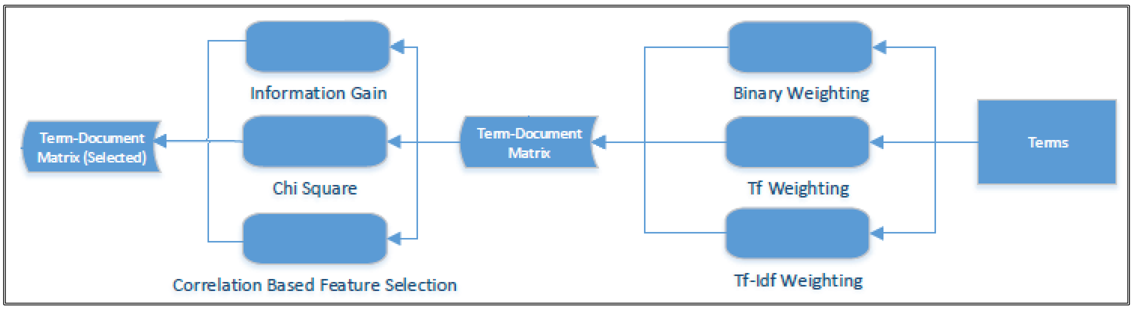


Figure 3.6. Term-Document matrix creation section

To begin with, terms are transformed into suitable form for text classifiers. Terms are transformed into an appropriate document representation model. Training data is represented as a set of feature vectors. Feature vector which demonstrates a document, includes one attribute for each word that exists in the dataset. If a feature exists in a document, then its related attribute is set to its occurrence number or any other calculated value. Consequently, each document is represented by the set of features by creating the term-document matrices. In term-document matrix, each row corresponds to a term and each column corresponds to a document. There are several different algorithms to make a decision of each record’s value in the term-document matrix. These values have important role in NLP.

In term-document matrix creation section, different weighting and feature selection algorithms are applied to generated terms which are the output of the previous section.

The algorithms are applied by using the machine learning software package called WEKA (Frenk and Witten, 2005). Weka is an open source data mining tool which is written in Java and also has many libraries in Java. WEKA supports filtering the features of data and computing weights of each feature in the document collection. WEKA includes a lot of machine learning algorithms for data mining jobs. Data preprocessing steps and classification tasks can be implemented with WEKA.

WEKA stores each set of data in a specific format. This format is called as *arff* file. WEKA can easily operate on the *arff* files. A database which is representing documents, feature vectors of documents, the values of the feature vector and the class labels of the documents are stored in this *arff* format.

3 different weighting algorithms, BW, TF and TF-IDF are applied. In the first stage, the classifier algorithms are directly applied to the feature vectors which are weighted with these algorithms. In the second stage, again 3 different feature selection algorithms which are called as X^2 statistic, IG and CBFS are applied to the weighted feature vectors. This means, 3 different selection algorithms are carried out on 3 different weighted feature vectors. Each selection algorithm is performed on 3 different weighted term-document matrices. 9 different combinations of weighting and feature selection algorithms are created. Totally, 12 different type feature vectors created. Table 3.2 is as following:

Table 3.2. 12 Different type feature vectors

Weighting Algorithms	Binary
	TF
	TF-IDF
Weighting + Feature Selection Algorithms	Binary + IG
	TF + IG
	TF-IDF + IG
	Binary + X^2
	TF + X^2
	TF-IDF + X^2
	Binary + CBFS
	TF + CBFS
	TF-IDF + CBFS

In the previous section which is called as term preparation, 6 different kinds of terms are generated. They are original, original + bi-gram, original + tri-gram, stem, stem + bi-gram, stem + tri-gram forms. 12 different types of feature vectors are created for 6 different kinds of terms. As a result, 72 different feature vectors are prepared to be classified by different classifier algorithms. 72 different feature vectors are constructed and saved as an *arff* file.

To begin with, weighting algorithms are applied for 6 different kinds of features. Original word representation has the highest attribute number. As seen in Table 3.3, the attribute number for BW, TF and TF-IDF weighting is the same, because weighting algorithms assigns weights to each attribute in the collection based on their occurrences. Weighting algorithms do not process any elimination on the attribute set based on the importance of the attribute. However, feature selection algorithms remove redundant and unnecessary noisy attributes. It is clearly seen from Table 3.3 that; CBFS algorithm removes more attributes than IG and X^2 statistic. Interestingly, IG and X^2 statistic eliminate the same number of attributes from the attribute set. To illustrate, the attribute number for stemmed words is 3028. IG algorithm reduces the number to 2977 and X^2 statistic also reduces the attribute number to 2977 for 3 weighting algorithms. This condition is the same for the other 5 types of features. The attribute numbers after implementing the feature selection algorithms IG and X^2 statistic are equal. But, the selected attributes are not known and the discriminative power of the attributes for each class is uncertain. It will be understood in the next section depending on the classification accuracy. Then, which feature selection algorithm chooses the best characteristic attributes for classification will be observed.

IG and X^2 statistic show the same performance while reducing the dimension. Moreover, CBFS shows better performance than IG and X^2 statistic. For instance, the attribute number for original words is 3945. IG and X^2 statistic reduce the attribute number to 3653, but CBFS reduces to 61 for binary weighting and to 99 for other weighting algorithms. Too much reduction may yield to loss of important data in dimension reduction process. While reducing the attributes, informative features can also be eliminated. Classification results will have an important role in determining the effectiveness of these feature selection algorithms in the next chapter.

Table 3.3. Attribute numbers after applying weighting and feature selection algorithms

	Original	Original + Bi-Gram	Original + Tri-Gram	Stem	Stem + Bi-Gram	Stem + Tri-Gram
BW	3945	894	1861	3028	894	2122
TF	3945	894	1861	3028	894	2122
TF-IDF	3945	894	1861	3028	894	2122
BW + IG	3657	730	1856	2977	728	2111
TF + IG	3653	729	1857	2977	727	2114
TF-IDF + IG	3653	729	1857	2977	727	2114
BW + CBFS	61	32	49	64	31	65
TF + CBFS	99	47	91	85	45	116
TF-IDF + CBFS	99	47	91	85	45	116
BW + X^2	3657	730	1856	2977	728	2111
TF + X^2	3653	729	1857	2977	727	2114
TF-IDF + X^2	3653	729	1857	2977	727	2114

With the completion of this term-document matrix creation section, all the preprocessing tasks are finished. All the files are ready to be classified in the next section.

3.4. Training and Testing

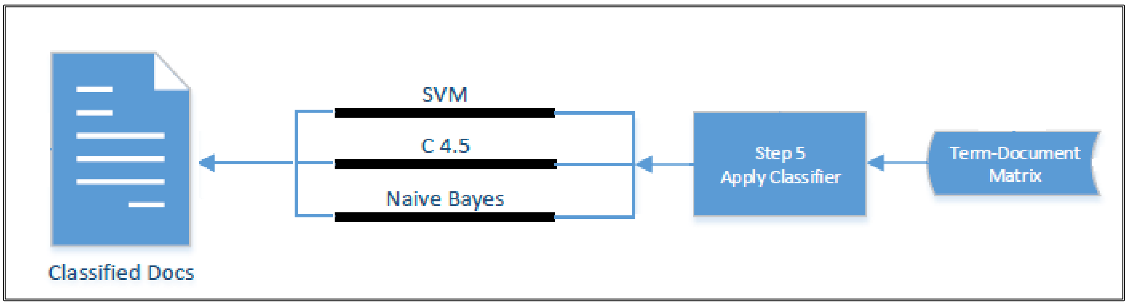


Figure 3.7. Training and testing section

The last and the most important section in document classification process is training and testing. In the previous section, all the term-document matrices are created. Now, all the features are in suitable form for text classifiers.

72 different feature vectors which are saved as an *arff* file are classified with 3 different classification algorithms. As seen in Figure 3.7, SVM, C4.5 and NB algorithms are implemented for the experiments. Default parameters in WEKA are used in all classification algorithms.

Initially, the classifier is trained with the entire dataset. In the training part, each classifier constructs the model to make predictions. Each classifier works on 57.553 instances to build the model. After completing the model construction, testing part starts. The algorithm predicts the classes of instances based on the model. The fundamental purpose of the thesis is to analyze the performance of different classification approaches which are composed from several algorithms. For this reason, the same conditions should be provided for all algorithms. Learning and testing of randomly selected datasets would create different environments and conditions for the algorithms. Thus, the whole dataset was used for both learning and testing. In testing part, test dataset is read and each instance is assigned to a predefined class. After completing the implementation of the classifiers, totally 216 results are collected for evaluation and discussion.

3.5. Summary

Document classification process has many stages. Preprocessing steps have crucial role in classification. Before starting experiments, all upper cases are transformed to lower cases, punctuation marks and stop words are removed and UTF-8 is used for character encoding. Dataset is prepared to an appropriate form for preprocessing steps. In this thesis, all the document representation methods are saved as *arrf* files in order to be used with WEKA program. Shortly, 3 weighting and 3 feature selection algorithms and the combination of weighting and feature selection algorithms are applied. Consequently, 12 different *arff* files for 6 different types of features (12*6) = 72 are created. These 72 *arff* files are used for classification. 3 different classification algorithms are implemented for 72 *arff* files (72 * 3) = 216. In total, 216 result files are gathered from WEKA.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1. Evaluation Metrics

Evaluation is a crucial and a challenging task in IR to design and implement an effective retrieval system. The main goal of the evaluation is to improve the system based on the results of evaluation metrics. Evaluation ensures future performance and success of the system.

There are several algorithms and systems in IR area; therefore decision to choose one of these algorithm and systems, then observing the performance of them depends on the results that evaluation metrics provide. Evaluation metrics show the relevancy of the retrieved documents.

In this thesis, 3 different classifiers are applied. Effectiveness of each classifier is measured by calculating the evaluation metrics, accuracy, recall, precision and F-measure.

4.1.1. Precision

Precision is one of the most frequent and basic measure for IR effectiveness. It shows the ability of a system to represent only relevant documents. Basically, precision is the number of relevant documents retrieved divided by the total number of documents retrieved.

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

A true positive result is detecting the condition when the condition is present. A true negative result is not detecting the condition when the condition is absent. A false positive result is detecting the condition when the condition is absent. A false negative result is not detecting the condition when the condition is present.

Table 4.1. TP, FP, FN, TN conditions

		Condition	
		Present	Absent
Result	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Table 4.1 can be easily summarized as follows:

Table 4.2. Relevant and non-relevant document conditions

	Relevant	Non-relevant
Retrieved	True Positives	False Positives
Not Retrieved	False Negative	True Negative

$$P = \frac{TP}{TP + FP} \tag{4.2}$$

4.1.2. Recall

Recall indicates the ability of a system to represent all relevant documents. Recall is the number of relevant documents retrieved divided by the total number of relevant documents.

According to the results from Table 4.2:

$$R = \frac{TP}{TP + FN} \tag{4.3}$$

Precision and recall are interdependent measures. Precision value generally decreases while the number of classified documents increases. On the other hand, recall value increases while the number of classified document increases.

4.1.3. F-Measure

In classification, F-measure is frequently used the standard evaluation metric. This metric is used to assess the performance of document and query classification. The F-measure is the harmonic mean of the precision and recall metrics.

$$F - Measure = \frac{2. Precision. Recall}{Precision + Recall} \tag{4.4}$$

F-measure value can take any value between 0 and 1. The value 0 indicates the poorest result which means no documents are classified correctly and the value 1 indicates a perfect result.

4.1.4. Accuracy

Accuracy is a well-liked evaluation measure in machine learning. Accuracy is the fraction of its classifications which are correct. Basically, this evaluation measure shows the closeness of a measured value is to the actual value.

$$Accuracy = \frac{True\ Positive(TP) + True\ Negative(TN)}{TP + TN + False\ Positive(FP) + False\ Negative(FN)} \qquad (4.5)$$

4.2. Results

In this part, performance of all different combinations for classification process will be presented. 216 different classification results are collected. 57.553 documents are classified using 3 different classification algorithms called NB, C4.5 and SVM. Then, 3 different weighting algorithms are applied; which are BW, TF and TF-IDF. 3 feature selection algorithms IG, X^2 statistic, CBFS and their combinations with weighting algorithms are implemented. Table 4.3 shows the results of all the algorithms applied during classification process. The accuracy values are given in Table 4.3. Throughout this chapter, the classification results are given with accuracy values. The classification results with other evaluation metrics such as recall, precision, F-measure, total number of correctly classified instances and total number of incorrectly classified instances are given in detail in Appendix B for further analysis.

Table 4.3 is a summary of all the results. Basically, it is divided into 3 columns. The top 3 columns denote the weighting algorithms. Each weighting algorithm is also divided into 3 columns which represent the classification algorithms. In principle, the table is split into 2 rows. Each row represents the form of the features, original and stem. Then, each row is partitioned into 12 rows. Every row shows how the document is represented and which feature selection algorithm is applied. To give an example, the accuracy value that is written bold is representing that TF weighting algorithm is applied with CBFS feature selection algorithm. Original word form and bi-gram document representation is used to apply the NB classifier. To give another example, the accuracy value that is written underlined and bold is representing that BW weighting algorithm is applied with no feature selection algorithm. Stemmed word form is used to apply the C4.5 classifier.

Table 4.3. Accuracy values of all the algorithms applied during classification process

		BW			**TF**			**TF-IDF**		
		NB	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**
Original	**No feature selection**	68.23	92.39	93.60	70.78	92.71	90.13	70.80	92.71	90.14
	IG	68.24	92.52	93.14	72.40	92.84	89.76	72.39	92.84	89.76
	X^2	68.24	92.44	93.14	72.30	92.88	89.77	72.30	92.88	89.77
	CBFS	65.43	71.49	67.32	64.09	81.05	71.41	64.09	81.05	71.42
	Bi-gram	47.44	88.89	78.46	47.22	91.19	80.40	47.25	91.19	80.39
	Bi-gram + IG	47.44	88.98	78.19	47.62	91.30	80.18	47.63	91.30	80.20
	Bi-gram + X^2	47.44	89.04	78.20	47.61	91.24	80.19	47.62	91.24	80.19
	Bi-gram + CBFS	50.34	61.81	53.45	**49.51**	78.89	60.13	49.48	78.89	60.13
	Tri-gram	54.73	92.49	89.81	58.97	93.51	87.97	58.96	93.51	87.97
	Tri-gram + IG	54.73	92.45	89.81	59.18	93.48	87.97	59.17	93.48	87.97
	Tri-gram + X^2	54.73	92.47	89.81	59.15	93.53	87.97	59.16	93.53	87.97
	Tri-gram + CBFS	62.87	72.23	66.08	60.34	85.10	71.20	60.34	85.10	71.20
Stem	**No feature selection**	68.79	**92.47**	92.34	71.96	92.93	89.13	72.00	92.93	89.13
	IG	68.79	92.49	92.34	73.01	92.91	89.04	73.04	92.91	89.04
	X^2	68.79	92.47	92.34	72.95	92.90	89.04	72.99	92.90	89.03
	CBFS	67.22	75.52	70.36	62.30	81.39	72.26	62.30	81.39	72.25
	Bi-gram	48.54	89.04	78.57	47.43	91.19	80.36	47.49	91.19	80.37
	Bi-gram + IG	48.55	89.11	78.37	47.83	91.34	80.22	47.87	91.34	80.21
	Bi-gram + X^2	48.55	89.08	78.36	47.83	91.35	80.22	47.87	91.35	80.20
	Bi-gram + CBFS	51.45	63.20	54.83	50.31	78.47	59.87	50.30	78.47	59.89
	Tri-gram	57.68	92.38	90.77	61.60	93.38	88.34	61.65	93.38	88.34
	Tri-gram + IG	57.69	92.28	90.77	61.86	93.40	88.34	61.90	93.40	88.33
	Tri-gram + X^2	57.69	92.30	90.77	61.82	93.53	88.34	61.82	93.53	88.34
	Tri-gram + CBFS	63.56	78.63	68.60	62.32	86.40	73.10	62.32	86.40	73.09

Table 4.3 is the big picture of all the results. In the following parts, Table 4.3 will be divided into smaller tables and will be examined deeply.

4.3. Evaluation and Discussion

In this thesis, we worked on Turkish news to classify the documents. 216 classification results are gathered with different combinations of weighting and feature selection algorithms.

Firstly, the 6 different kinds of feature types are examined. Table 4.4 shows the experimental results of 6 different feature types with weighting algorithms. To see which feature type gives the best performance with only weighting algorithms, the results are viewed vertically. The values highlighted with bold indicate the best performance for each classifier. It is clearly seen from Table 4.4 that no matter what weighting algorithm is applied, NB always outputs the best result for stemmed words, C4.5 always outputs the best result for original words which are represented with tri-gram and SVM always outputs the best result for original words. But, the results change depending on the weighting algorithm. SVM performs the best with BW, NB performs the best with TF-IDF and C4.5 performs the same with TF and TF-IDF, however their accuracy value is better than BW.

Table 4.4. Experimental results of 6 different feature types with weighting algorithms to see which feature type gives the best performance

		BW			**TF**			**TF-IDF**		
		NB	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**
Original	**No feature selection**	68.23	92.39	**93.60**	70.78	92.71	**90.13**	70.80	92.71	**90.14**
	Bi-gram	47.44	88.89	78.46	47.22	91.19	80.40	47.25	91.19	80.39
	Tri-gram	54.73	**92.49**	89.81	58.97	**93.51**	87.97	58.96	**93.51**	87.97
Stem	**No feature selection**	**68.79**	92.47	92.34	**71.96**	92.93	89.13	**72.00**	92.93	89.13
	Bi-gram	48.54	89.04	78.57	47.43	91.19	80.36	47.49	91.19	80.37
	Tri-gram	57.68	92.38	90.77	61.60	93.38	88.34	61.65	93.38	88.34

Secondly, Table 4.5 shows the experimental results of 6 different feature types with weighting algorithms. To see which classifier gives the best performance with only weighting, algorithms, the results are viewed horizontally. The values highlighted with bold indicate the best performance for each row. In the first row, SVM gives the best result for original word form weighted with BW. But, in other rows, the best classifier is C4.5. The accuracy value for C4.5 is the same with TF and TF-IDF. This means, there is no difference between using the TF and TF-IDF weighting algorithms for C4.5 for any feature representation types. C4.5 gives the best result for 5 feature representation types.

Table 4.5. Experimental results of 6 different feature types with weighting algorithms to see which classifier gives the best performance

		BW			**TF**			**TF-IDF**		
		NB	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**
Original	**No feature selection**	68.23	92.39	**93.60**	70.78	92.71	90.13	70.80	92.71	90.14
	Bi-gram	47.44	88.89	78.46	47.22	**91.19**	80.40	47.25	**91.19**	80.39
	Tri-gram	54.73	92.49	89.81	58.97	**93.51**	87.97	58.96	**93.51**	87.97
Stem	**No feature selection**	68.79	92.47	92.34	71.96	**92.93**	89.13	72.00	**92.93**	89.13
	Bi-gram	48.54	89.04	78.57	47.43	**91.19**	80.36	47.49	**91.19**	80.37
	Tri-gram	57.68	92.38	90.77	61.60	**93.38**	88.34	61.65	**93.38**	88.34

The feature selection algorithms are not considered in the above tables. Table 4.6 shows the results after applying the feature selection algorithms. To see which feature type gives the best performance with the combination of weighting and feature selection algorithms, the results are viewed vertically. The values highlighted with bold indicate the best performance for each classifier.

NB always performs the best for stemmed words. IG feature selection algorithm accomplishes the best result for 3 weighting algorithms. Additionally with BW, X^2 feature selection algorithm outputs the same accuracy value with IG algorithm.

C4.5 achieves the best result twice for original and twice for stemmed words with tri-gram representation and TF or TF-IDF weighting with X^2 statistic combination. The accuracy value is equal to %93.53 for 4 results. Also, C4.5 achieves the best result for original words and BW plus IG combination. But, the accuracy value is equal to %92.25 with BW that is lower than TF and TF-IDF weighting.

SVM performs the best for original words with X^2 statistic feature selection algorithm, no matter which weighting algorithm is applied. Only for BW, the accuracy values are equal to each other with IG and X^2 feature selection algorithms.

Table 4.6. Results after applying the feature selection algorithms to see which feature type gives the best performance

		BW			**TF**			**TF-IDF**		
		NB	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**
Original	**IG**	68.24	**92.52**	**93.14**	72.40	92.84	89.76	72.39	92.84	89.76
	X^2	68.24	92.44	**93.14**	72.30	92.88	**89.77**	72.30	92.88	**89.77**
	CBFS	65.43	71.49	67.32	64.09	81.05	71.41	64.09	81.05	71.42
	Bi-gram + IG	47.44	88.98	78.19	47.62	91.30	80.18	47.63	91.30	80.20
	Bi-gram + X^2	47.44	89.04	78.20	47.61	91.24	80.19	47.62	91.24	80.19
	Bi-gram + CBFS	50.34	61.81	53.45	49.51	78.89	60.13	49.48	78.89	60.13
	Tri-gram + IG	54.73	92.45	89.81	59.18	93.48	87.97	59.17	93.48	87.97
	Tri-gram + X^2	54.73	92.47	89.81	59.15	**93.53**	87.97	59.16	**93.53**	87.97
	Tri-gram + CBFS	62.87	72.23	66.08	60.34	85.10	71.20	60.34	85.10	71.20
Stem	**IG**	**68.79**	92.49	92.34	**73.01**	92.91	89.04	**73.04**	92.91	89.04
	X^2	**68.79**	92.47	92.34	72.95	92.90	89.04	72.99	92.90	89.03
	CBFS	67.22	75.52	70.36	62.30	81.39	72.26	62.30	81.39	72.25
	Bi-gram + IG	48.55	89.11	78.37	47.83	91.34	80.22	47.87	91.34	80.21
	Bi-gram + X^2	48.55	89.08	78.36	47.83	91.35	80.22	47.87	91.35	80.20
	Bi-gram + CBFS	51.45	63.20	54.83	50.31	78.47	59.87	50.30	78.47	59.89
	Tri-gram + IG	57.69	92.28	90.77	61.86	93.40	88.34	61.90	93.40	88.33
	Tri-gram + X^2	57.69	92.30	90.77	61.82	**93.53**	88.34	61.82	**93.53**	88.34
	Tri-gram + CBFS	63.56	78.63	68.60	62.32	86.40	73.10	62.32	86.40	73.09

Again, Table 4.7 shows the experimental results of 6 different feature types with weighting and feature selection algorithms. To see which classifier gives the best performance with the combination of weighting and feature selection algorithms, the results are viewed horizontally. The values emphasized with bold indicate the best performance for each row.

Table 4.7. Experimental results of 6 different feature types with weighting and feature selection algorithms to see which classifier gives the best performance

		BW			**TF**			**TF-IDF**		
		NB	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**
Original	**IG**	68.24	92.52	**93.14**	72.40	92.84	89.76	72.39	92.84	89.76
	X^2	68.24	92.44	**93.14**	72.30	92.88	89.77	72.30	92.88	89.77
	CBFS	65.43	71.49	67.32	64.09	**81.05**	71.41	64.09	**81.05**	71.42
	Bi-gram + IG	47.44	88.98	78.19	47.62	**91.30**	80.18	47.63	**91.30**	80.20
	Bi-gram + X^2	47.44	89.04	78.20	47.61	**91.24**	80.19	47.62	**91.24**	80.19
	Bi-gram + CBFS	50.34	61.81	53.45	49.51	**78.89**	60.13	49.48	**78.89**	60.13
	Tri-gram + IG	54.73	92.45	89.81	59.18	**93.48**	87.97	59.17	**93.48**	87.97
	Tri-gram + X^2	54.73	92.47	89.81	59.15	**93.53**	87.97	59.16	**93.53**	87.97
	Tri-gram + CBFS	62.87	72.23	66.08	60.34	**85.10**	71.20	60.34	**85.10**	71.20
Stem	**IG**	68.79	92.49	92.34	73.01	**92.91**	89.04	73.04	**92.91**	89.04
	X^2	68.79	92.47	92.34	72.95	**92.90**	89.04	72.99	**92.90**	89.03
	CBFS	67.22	75.52	70.36	62.30	**81.39**	72.26	62.30	**81.39**	72.25
	Bi-gram + IG	48.55	89.11	78.37	47.83	**91.34**	80.22	47.87	**91.34**	80.21
	Bi-gram + X^2	48.55	89.08	78.36	47.83	**91.35**	80.22	47.87	**91.35**	80.20
	Bi-gram + CBFS	51.45	63.20	54.83	50.31	**78.47**	59.87	50.30	**78.47**	59.89
	Tri-gram + IG	57.69	92.28	90.77	61.86	**93.40**	88.34	61.90	**93.40**	88.33
	Tri-gram + X^2	57.69	92.30	90.77	61.82	**93.53**	88.34	61.82	**93.53**	88.34
	Tri-gram + CBFS	63.56	78.63	68.60	62.32	**86.40**	73.10	62.32	**86.40**	73.09

SVM outputs the best accuracy value for only original word and BW plus IG or BW plus X^2 statistic combinations. Two results are equal. Furthermore, for the other results, C4.5 always have the highest accuracy value for all combinations of TF and TF-IDF weighting with feature selection algorithms.

Next, the effect of feature selection algorithms is taken into consideration. The following 6 different tables show the best feature selection algorithms. The highest accuracy values for different combinations of weighting and feature selection algorithms for 6 different feature representation types can be seen from the following 6 tables, namely Table 4.8, Table 4.9, Table 4.10, Table 4.11, Table 4.12 and Table 4.13.

Table 4.8. The best feature selection algorithms for original words

	Original								
	BW			**TF**			**TF-IDF**		
	NB	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**
IG	**68.24**	**92.52**	**93.14**	**72.40**	92.84	89.76	**72.39**	92.84	89.76
X^2	**68.24**	92.44	**93.14**	72.30	**92.88**	**89.77**	72.30	**92.88**	**89.77**
CBFS	65.43	71.49	67.32	64.09	81.05	71.41	64.09	81.05	71.42

Table 4.9. The best feature selection algorithms for original words with bi-gram document representation

	Original								
	BW			**TF**			**TF-IDF**		
	NB	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**
Bi-gram + IG	**47.44**	**88.98**	78.19	47.62	**91.30**	80.18	47.63	**91.30**	**80.20**
Bi-gram + X^2	**47.44**	89.04	**78.20**	47.61	91.24	**80.19**	47.62	91.24	80.19
Bi-gram + CBFS	50.34	61.81	53.45	**49.51**	78.89	60.13	**49.48**	78.89	60.13

Table 4.10. The best feature selection algorithms for original words with tri-gram document representation

	Original								
	BW			**TF**			**TF-IDF**		
	NB	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**
Tri-gram + IG	54.73	92.45	**89.81**	59.18	93.48	**87.97**	59.17	93.48	**87.97**
Tri-gram + X^2	54.73	**92.47**	**89.81**	59.15	**93.53**	**87.97**	59.16	**93.53**	**87.97**
Tri-gram + CBFS	**62.87**	72.23	66.08	**60.34**	85.10	71.20	**60.34**	85.10	71.20

Table 4.11. The best feature selection algorithms for stemmed words

	Stem								
	BW			**TF**			**TF-IDF**		
	NB	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**
IG	**68.79**	**92.49**	**92.34**	**73.01**	**92.91**	**89.04**	**73.04**	**92.91**	**89.04**
X^2	**68.79**	92.47	**92.34**	72.95	92.90	**89.04**	72.99	92.90	89.03
CBFS	67.22	75.52	70.36	62.30	81.39	72.26	62.30	81.39	72.25

Table 4.12. The best feature selection algorithms for stemmed words with bi-gram document representation

	Stem								
	BW			**TF**			**TF-IDF**		
	NB	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**
Bi-gram + IG	48.55	**89.11**	**78.37**	47.83	91.34	**80.22**	47.87	91.34	**80.21**
Bi-gram + X^2	48.55	89.08	78.36	47.83	**91.35**	**80.22**	47.87	**91.35**	80.20
Bi-gram + CBFS	**51.45**	63.20	54.83	**50.31**	78.47	59.87	**50.30**	78.47	59.89

Table 4.13. The best feature selection algorithms for stemmed words with tri-gram document representation

	Stem								
	BW			**TF**			**TF-IDF**		
	NB	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**
Tri-gram + IG	57.69	92.28	**90.77**	**61.86**	93.40	**88.34**	61.90	93.40	88.33
Tri-gram + X^2	57.69	**92.30**	**90.77**	61.82	**93.53**	**88.34**	61.82	**93.53**	**88.34**
Tri-gram + CBFS	**63.56**	78.63	68.60	62.32	86.40	73.10	**62.32**	86.40	73.09

In the previous chapter**,** from Table 3.3, it is observed that IG and X^2 statistic algorithms reduce the same number of attributes. But the classification results which are applied to these attributes selected by IG and X^2 statistic algorithms are not the same. The

NB accuracy results are always the same for the IG and X^2 statistic combinations. Implementing the IG or X^2 statistic for NB classification does not change the result in any representation types. In addition, from Table 3.3, it is indicated that CBFS algorithm reduces the highest number of attributes from the original attribute set. But, it can be observed that reducing the highest number of attributes does not make the CBFS algorithm the best feature selection algorithm. Because the accuracy values of CBFS algorithm is generally lower than the IG and X^2 statistic algorithms. Only 10 times in 162 results listed in above 6 tables, the CBFS algorithm performs better.

Then, the next 3 tables are separated based on the applied weighting algorithm. Table 4.14 shows the accuracy values for BW and different combinations of implemented algorithms. With BW, NB algorithm has the highest accuracy value for stemmed words with no feature selection algorithm and also has the highest accuracy value for stemmed words with IG feature selection algorithm. In this case, the combination with no feature selection algorithm can be selected. Because their accuracy values are equal and then there is no need to apply an extra selection algorithm. C4.5 has the highest accuracy value for original words with IG feature selection algorithm and SVM has the highest accuracy for original words with no feature selection algorithm.

Table 4.14. Accuracy values for BW and different combinations of implemented algorithms

	BW					
	Original			Stem		
	NB	C4.5	SVM	NB	C4.5	SVM
No feature selection	68.23	92.39	**93.60**	**68.79**	92.47	92.34
IG	68.24	**92.52**	93.14	**68.79**	92.49	92.34
X^2	68.24	92.44	93.14	68.79	92.47	92.34
CBFS	65.43	71.49	67.32	67.22	75.52	70.36
Bi-gram	47.44	88.89	78.46	48.54	89.04	78.57
Bi-gram + IG	47.44	88.98	78.19	48.55	89.11	78.37
Bi-gram + X^2	47.44	89.04	78.20	48.55	89.08	78.36
Bi-gram + CBFS	50.34	61.81	53.45	51.45	63.20	54.83
Tri-gram	54.73	92.49	89.81	57.68	92.38	90.77
Tri-gram + IG	54.73	92.45	89.81	57.69	92.28	90.77
Tri-gram + X^2	54.73	92.47	89.81	57.69	92.30	90.77
Tri-gram + CBFS	62.87	72.23	66.08	63.56	78.63	68.60

Table 4.15 shows the accuracy values for TF and different combinations of implemented algorithms. With TF, NB algorithm has the highest accuracy for stemmed words with IG feature selection algorithm, C4.5 has the highest accuracy for original words with tri-gram representation and X^2 statistic feature selection algorithm. SVM has the highest accuracy for original words with no feature selection algorithm. It is seen from Table 4.15 that there is no need to apply any preprocessing steps for SVM classifier to get the best result.

Table 4.15. Accuracy values for TF and different combinations of implemented algorithms

	TF					
	Original			Stem		
	NB	C4.5	SVM	NB	C4.5	SVM
No feature selection	70.78	92.71	**90.13**	71.96	92.93	89.13
IG	72.40	92.84	89.76	**73.01**	92.91	89.04
X^2	72.30	92.88	89.77	72.95	92.90	89.04
CBFS	64.09	81.05	71.41	62.30	81.39	72.26
Bi-gram	47.22	91.19	80.40	47.43	91.19	80.36
Bi-gram + IG	47.62	91.30	80.18	47.83	91.34	80.22
Bi-gram + X^2	47.61	91.24	80.19	47.83	91.35	80.22
Bi-gram + CBFS	49.51	78.89	60.13	50.31	78.47	59.87
Tri-gram	58.97	93.51	87.97	61.60	93.38	88.34
Tri-gram + IG	59.18	93.48	87.97	61.86	93.40	88.34
Tri-gram + X^2	59.15	**93.53**	87.97	61.82	93.53	88.34
Tri-gram + CBFS	60.34	85.10	71.20	62.32	86.40	73.10

Table 4.16 shows the accuracy values for TF-IDF and different combinations of implemented algorithms. With TF-IDF, NB algorithm has the highest accuracy for stemmed words with X^2 statistic feature selection algorithm, C4.5 has the highest accuracy for original words with tri-gram representation and X^2 statistic feature selection algorithm. Also, SVM has the highest accuracy for original words with no feature selection algorithm.

Table 4.16. Accuracy values for TF-IDF and different combinations of implemented algorithms

	TF-IDF					
	Original			**Stem**		
	NB	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**
No feature selection	70.80	92.71	**90.14**	72.00	92.93	89.13
IG	72.39	92.84	89.76	73.04	92.91	89.04
X^2	72.30	92.88	89.77	**72.99**	92.90	89.03
CBFS	64.09	81.05	71.42	62.30	81.39	72.25
Bi-gram	47.25	91.19	80.39	47.49	91.19	80.37
Bi-gram + IG	47.63	91.30	80.20	47.87	91.34	80.21
Bi-gram + X^2	47.62	91.24	80.19	47.87	91.35	80.20
Bi-gram + CBFS	49.48	78.89	60.13	50.30	78.47	59.89
Tri-gram	58.96	93.51	87.97	61.65	93.38	88.34
Tri-gram + IG	59.17	93.48	87.97	61.90	93.40	88.33
Tri-gram + X^2	59.16	**93.53**	87.97	61.82	93.53	88.34
Tri-gram + CBFS	60.34	85.10	71.20	62.32	86.40	73.09

There is no doubt that SVM classifier always performs the best when the original words are used with no feature selection algorithm. To compare 3 accuracy values, the highest one is written underlined which is %93.60 with BW as shown in Table 4.17.

Table 4.17. Performance of SVM classifier when the original words used with no feature selection algorithm

SVM		
Original		
BW	**TF**	**TF-IDF**
93.60	**90.13**	**90.14**

NB algorithm has the highest accuracy written underlined with %73.04 for stemmed words with TF-IDF weighting and IG feature selection algorithm as shown in Table 4.18.

Table 4.18. Performance of NB classifier when stemmed words used

	NB		
	Stem		
	BW	**TF**	**TF-IDF**
No feature selection	**68.79**	71.96	72.00
IG	**68.79**	**73.01**	**73.04**
X^2	**68.79**	72.95	72.99
CBFS	67.22	62.30	62.30
Bi-gram	48.54	47.43	47.49
Bi-gram + IG	48.55	47.83	47.87
Bi-gram + X^2	48.55	47.83	47.87
Bi-gram + CBFS	51.45	50.31	50.30
Tri-gram	57.68	61.60	61.65
Tri-gram + IG	57.69	61.86	61.90
Tri-gram + X^2	57.69	61.82	61.82
Tri-gram + CBFS	63.56	62.32	62.32

C4.5 always has the same accuracy values for all the combinations when TF and TF-IDF weighting algorithms are applied as shown in Table 4.19. C4.5 has the highest accuracy value written underlined as %93.53 for original words with tri-gram representation and X^2 statistic feature selection algorithm and TF or TF-IDF weighting algorithm.

Table 4.19. Performance of C4.5 classifier when original words used for all combinations

	C4.5		
	Original		
	BW	**TF**	**TF-IDF**
No feature selection	92.39	92.71	92.71
IG	**92.52**	92.84	92.84
X^2	92.44	92.88	92.88
CBFS	71.49	81.05	81.05
Bi-gram	88.89	91.19	91.19
Bi-gram + IG	88.98	91.30	91.30
Bi-gram + X^2	89.04	91.24	91.24
Bi-gram + CBFS	61.81	78.89	78.89
Tri-gram	92.49	93.51	93.51
Tri-gram + IG	92.45	93.48	93.48
Tri-gram + X^2	92.47	**93.53**	**93.53**
Tri-gram + CBFS	72.23	85.10	85.10

To summarize, the next table is called as the big picture in the previous part. Table 4.20 shows 216 classification results. To see which classifier gives the best accuracy value with the combination of weighting and feature selection algorithms, the results are viewed vertically. The values highlighted with bold indicate the highest accuracy value for each classification algorithm.

Table 4.20. The highest accuracy values for each classification algorithm

		BW			**TF**			**TF-IDF**		
		NB	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**
Original	**No feature selection**	68.23	92.39	**93.60**	70.78	92.71	**90.13**	70.80	92.71	**90.14**
	IG	68.24	**92.52**	93.14	72.40	92.84	89.76	72.39	92.84	89.76
	X^2	68.24	92.44	93.14	72.30	92.88	89.77	72.30	92.88	89.77
	CBFS	65.43	71.49	67.32	64.09	81.05	71.41	64.09	81.05	71.42
	Bi-gram	47.44	88.89	78.46	47.22	91.19	80.40	47.25	91.19	80.39
	Bi-gram + IG	47.44	88.98	78.19	47.62	91.30	80.18	47.63	91.30	80.20
	Bi-gram + X^2	47.44	89.04	78.20	47.61	91.24	80.19	47.62	91.24	80.19
	Bi-gram + CBFS	50.34	61.81	53.45	49.51	78.89	60.13	49.48	78.89	60.13
	Tri-gram	54.73	92.49	89.81	58.97	93.51	87.97	58.96	93.51	87.97
	Tri-gram + IG	54.73	92.45	89.81	59.18	93.48	87.97	59.17	93.48	87.97
	Tri-gram + X^2	54.73	92.47	89.81	59.15	**93.53**	87.97	59.16	**93.53**	87.97
	Tri-gram + CBFS	62.87	72.23	66.08	60.34	85.10	71.20	60.34	85.10	71.20
Stem	**No feature selection**	**68.79**	92.47	92.34	71.96	92.93	89.13	72.00	92.93	89.13
	IG	**68.79**	92.49	92.34	**73.01**	92.91	89.04	**73.04**	92.91	89.04
	X^2	**68.79**	92.47	92.34	72.95	92.90	89.04	72.99	92.90	89.03
	CBFS	67.22	75.52	70.36	62.30	81.39	72.26	62.30	81.39	72.25
	Bi-gram	48.54	89.04	78.57	47.43	91.19	80.36	47.49	91.19	80.37
	Bi-gram + IG	48.55	89.11	78.37	47.83	91.34	80.22	47.87	91.34	80.21
	Bi-gram + X^2	48.55	89.08	78.36	47.83	91.35	80.22	47.87	91.35	80.20
	Bi-gram + CBFS	51.45	63.20	54.83	50.31	78.47	59.87	50.30	78.47	59.89
	Tri-gram	57.68	92.38	90.77	61.60	93.38	88.34	61.65	93.38	88.34
	Tri-gram + IG	57.69	92.28	90.77	61.86	93.40	88.34	61.90	93.40	88.33
	Tri-gram + X^2	57.69	92.30	90.77	61.82	**93.53**	88.34	61.82	**93.53**	88.34
	Tri-gram + CBFS	63.56	78.63	68.60	62.32	86.40	73.10	62.32	86.40	73.09

Table 4.21 shows the highest accuracy values for each combination, the results are viewed horizontally. The values highlighted with bold indicate the highest accuracy value for each row separated by weighting algorithm. For each weighting algorithm, C4.5 outputs the highest accuracy values except for the first 3 rows. In the first 3 rows, SVM outputs the highest accuracy values.

Table 4.21. The highest accuracy values for each combination

		BW			TF			TF-IDF		
		NB	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**
Original	**No feature selection**	68.23	92.39	**93.60**	70.78	**92.71**	90.13	70.80	**92.71**	90.14
	IG	68.24	92.52	**93.14**	72.40	**92.84**	89.76	72.39	**92.84**	89.76
	X^2	68.24	92.44	**93.14**	72.30	**92.88**	89.77	72.30	**92.88**	89.77
	CBFS	65.43	**71.49**	67.32	64.09	**81.05**	71.41	64.09	**81.05**	71.42
	Bi-gram	47.44	**88.89**	78.46	47.22	**91.19**	80.40	47.25	**91.19**	80.39
	Bi-gram + IG	47.44	**88.98**	78.19	47.62	**91.30**	80.18	47.63	**91.30**	80.20
	Bi-gram + X^2	47.44	**89.04**	78.20	47.61	**91.24**	80.19	47.62	**91.24**	80.19
	Bi-gram + CBFS	50.34	**61.81**	53.45	49.51	**78.89**	60.13	49.48	**78.89**	60.13
	Tri-gram	54.73	**92.49**	89.81	58.97	**93.51**	87.97	58.96	**93.51**	87.97
	Tri-gram + IG	54.73	**92.45**	89.81	59.18	**93.48**	87.97	59.17	**93.48**	87.97
	Tri-gram + X^2	54.73	**92.47**	89.81	59.15	**93.53**	87.97	59.16	**93.53**	87.97
	Tri-gram + CBFS	62.87	**72.23**	66.08	60.34	**85.10**	71.20	60.34	**85.10**	71.20
Stem	**No feature selection**	68.79	**92.47**	92.34	71.96	**92.93**	89.13	72.00	**92.93**	89.13
	IG	68.79	**92.49**	92.34	73.01	**92.91**	89.04	73.04	**92.91**	89.04
	X^2	68.79	**92.47**	92.34	72.95	**92.90**	89.04	72.99	**92.90**	89.03
	CBFS	67.22	**75.52**	70.36	62.30	**81.39**	72.26	62.30	**81.39**	72.25
	Bi-gram	48.54	**89.04**	78.57	47.43	**91.19**	80.36	47.49	**91.19**	80.37
	Bi-gram + IG	48.55	**89.11**	78.37	47.83	**91.34**	80.22	47.87	**91.34**	80.21
	Bi-gram + X^2	48.55	**89.08**	78.36	47.83	**91.35**	80.22	47.87	**91.35**	80.20
	Bi-gram + CBFS	51.45	**63.20**	54.83	50.31	**78.47**	59.87	50.30	**78.47**	59.89
	Tri-gram	57.68	**92.38**	90.77	61.60	**93.38**	88.34	61.65	**93.38**	88.34
	Tri-gram + IG	57.69	**92.28**	90.77	61.86	**93.40**	88.34	61.90	**93.40**	88.33
	Tri-gram + X^2	57.69	**92.30**	90.77	61.82	**93.53**	88.34	61.82	**93.53**	88.34
	Tri-gram + CBFS	63.56	**78.63**	68.60	62.32	**86.40**	73.10	62.32	**86.40**	73.09

To conclude, Table 4.22 depicts the highest accuracy values among the 216 findings, the results are viewed horizontally. The values highlighted with bold indicate the highest accuracy value for each row with no logical separation. Then, it can be clearly seen that the highest accuracy values belong to which classifier and combination.

As a consequence of that, SVM performs the best 3 times for original words weighted by BW with no feature selection algorithm and also with IG and X^2 statistic feature selection algorithms. In this situation, the combination with no feature selection for original words can be picked because its accuracy value is higher than the others. In all other cases, which are weighted by TF or TF-IDF, because two of them output the same accuracy values, C4.5 algorithm performs the best for Turkish News classification.

Table 4.22. The highest accuracy values among the 216 findings

		BW			**TF**			**TF-IDF**		
		NB	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**	**NB**	**C4.5**	**SVM**
Original	**No feature selection**	68.23	92.39	**93.60**	70.78	92.71	90.13	70.80	92.71	90.14
	IG	68.24	92.52	**93.14**	72.40	92.84	89.76	72.39	92.84	89.76
	X^2	68.24	92.44	**93.14**	72.30	92.88	89.77	72.30	92.88	89.77
	CBFS	65.43	71.49	67.32	64.09	**81.05**	71.41	64.09	**81.05**	71.42
	Bi-gram	47.44	88.89	78.46	47.22	**91.19**	80.40	47.25	**91.19**	80.39
	Bi-gram + IG	47.44	88.98	78.19	47.62	**91.30**	80.18	47.63	**91.30**	80.20
	Bi-gram + X^2	47.44	89.04	78.20	47.61	**91.24**	80.19	47.62	**91.24**	80.19
	Bi-gram + CBFS	50.34	61.81	53.45	49.51	**78.89**	60.13	49.48	**78.89**	60.13
	Tri-gram	54.73	92.49	89.81	58.97	**93.51**	87.97	58.96	**93.51**	87.97
	Tri-gram + IG	54.73	92.45	89.81	59.18	**93.48**	87.97	59.17	**93.48**	87.97
	Tri-gram + X^2	54.73	92.47	89.81	59.15	**93.53**	87.97	59.16	**93.53**	87.97
	Tri-gram + CBFS	62.87	72.23	66.08	60.34	**85.10**	71.20	60.34	**85.10**	71.20
Stem	**No feature selection**	68.79	92.47	92.34	71.96	**92.93**	89.13	72.00	**92.93**	89.13
	IG	68.79	92.49	92.34	73.01	**92.91**	89.04	73.04	**92.91**	89.04
	X^2	68.79	92.47	92.34	72.95	**92.90**	89.04	72.99	**92.90**	89.03
	CBFS	67.22	75.52	70.36	62.30	**81.39**	72.26	62.30	**81.39**	72.25
	Bi-gram	48.54	89.04	78.57	47.43	**91.19**	80.36	47.49	**91.19**	80.37
	Bi-gram + IG	48.55	89.11	78.37	47.83	**91.34**	80.22	47.87	**91.34**	80.21
	Bi-gram + X^2	48.55	89.08	78.36	47.83	**91.35**	80.22	47.87	**91.35**	80.20
	Bi-gram + CBFS	51.45	63.20	54.83	50.31	**78.47**	59.87	50.30	**78.47**	59.89
	Tri-gram	57.68	92.38	90.77	61.60	**93.38**	88.34	61.65	**93.38**	88.34
	Tri-gram + IG	57.69	92.28	90.77	61.86	**93.40**	88.34	61.90	**93.40**	88.33
	Tri-gram + X^2	57.69	92.30	90.77	61.82	**93.53**	88.34	61.82	**93.53**	88.34
	Tri-gram + CBFS	63.56	78.63	68.60	62.32	**86.40**	73.10	62.32	**86.40**	73.09

CHAPTER 5

CONCLUSION

Today, the number of electronic documents is growing very rapidly. The documents are unstructured. Getting the desired information from these unstructured documents is difficult. At this point, IR techniques come to light to access the valuable information quickly, to summarize the information, to classify the documents and many other solutions are produced.

Document classification is one of the important tasks in IR. A lot of documents are stored in the hand-created folders, depending on their topics. When the documents size goes up, managing and processing the documents turn into a challenging task for humans. Therefore, the need for classification systems arises. Document classification is considered very critical and important to manage and access the related information.

Generally, document classification is very popular in English. In this thesis, different document classification approaches are applied and evaluated on Turkish documents, so this situation makes the thesis important for text classification researches in Turkish. Milliyet Dataset is used in the classification process which is created from Turkish news. The dataset has 9 classes. The classes are “Dünya”, “Yaşam”, “Spor”, “Sanat”, “Güncel”, “Siyaset”, “Magazin”, “Ekonomi” and “Sağlık”. The data set is read from an XML file and then transformed into 57.553 different Turkish news.

After preparing the dataset, preprocessing steps are implemented. Stop words are eliminated from the documents. After removing the stop words, generating the terms phase starts. All the features in the documents are represented in 6 different forms. Original form of the word and stem of the word are taken. Then, these 2 forms are represented using n-grams. Bi-gram and tri-gram representations are used. Totally, 6 types of features exist.

With the completion of generating the terms, terms are weighted with 3 different term weighting algorithms, BW, TF and TF-IDF weighting. 3 term-document matrices are created with term weights. Moreover, 3 feature selection algorithms are applied to these weighted terms to eliminate the noisy and irrelevant terms. Next, feature selection algorithms choose the feature subsets which have the most discriminative power for classification from the attribute set. 9 different combinations come into being for 3 weighting and 3 feature selection algorithms. Thus, for 6 kinds of feature types, 12 term-document matrices are created. 72 arff files are generated in total by using the data mining tool WEKA.

After preprocessing steps, all the documents are appropriate for classifiers. 3 classification algorithms are used. The classification algorithms used in this study are SVM, C4.5 and NB. Primarily, 3 classification algorithms are directly applied to weighted terms so as to compare the effects of weighting algorithms on classification accuracy. Then, 3 classifiers are applied to combinations of the weighting and feature selection algorithms. Consequently, 216 classification results are generated for 72 prepared arff files.

Classifiers achieve the best results with different conditions. NB always outputs the best result for stemmed words, no matter what weighting algorithm is applied. C4.5 always outputs the best result for original words which are represented with tri-grams. SVM always outputs the best result for original words. But, the results change depending on the weighting algorithm. SVM performs the best with BW, NB performs the best with TF-IDF and C4.5 performs the same with TF and TF-IDF, however their accuracy is better than BW.

The accuracy value for C4.5 is the same with TF and TF-IDF. It is clearly seen that there is no difference between using the TF and TF-IDF weighting algorithms for C4.5 for any feature representation types. For 3 weighting algorithms, NB always performs the best with stemmed words and IG feature selection algorithm. SVM performs the best for original word with X^2 statistic feature selection algorithm no matter what weighting algorithm is applied. C4.5 always has the highest accuracy values for TF and TF-IDF weighting with X^2 statistic feature selection algorithm when tri-gram representation is used for both stem and original form of the word.

In short, SVM classifier always performs the best when the original words used with no feature selection algorithm. It is clearly seen that preprocessing steps has no positive effect on the success of SVM algorithm. C4.5 always has the same accuracy values for all combinations when TF and TF-IDF weighting algorithms are applied. NB has the highest accuracy with %73.04 for stemmed words with TF-IDF weighting plus IG feature selection algorithms. C4.5 has the highest accuracy value with %93.53 for original and stemmed words with tri-gram representation and X^2 statistic feature selection algorithm when TF or TF-IDF weighting algorithm is applied.

As a consequence of that, SVM performs the best 3 times for original words weighted by BW when IG or X^2 statistic or no feature selection algorithms are applied. In all other cases which are weighted by TF or TF-IDF, C4.5 algorithm performs the best for Turkish News classification.

When everything is taken into account, experimental results indicate that C4.5 classification algorithm has the highest accuracy in 95% of the results. SVM provides the highest accuracy in 5% of the results. NB algorithm has always the lowest accuracy rate among 3 different classification algorithm results. After all, it can be stated that the most accurate results are obtained using C4.5 classification algorithm for Turkish News. The accuracy values of C4.5 algorithm with TF and TF-IDF weighting algorithms are the same. Based on these results, it can be concluded that there is no need to calculate IDF values to find TF-IDF value. In conclusion, C4.5 algorithm with TF weighting algorithm gives the best results for Turkish News.

In future, some other term weighting, feature selection and stemming algorithms can be examined. 3 different classification algorithms are applied in this thesis; in future, other classification algorithms can be applied individually and in combinations. Also, semantic algorithms can be implemented to represent the documents as a set of concepts rather than as a set of individual words for text classification.

REFERENCES

Aggarwal C. C., & Zhai C. (2012). Mining Text Data. Springer-Verlag New York Inc.

Akın, A. A., & Akın, M. D. (2007). Zemberek, an Open Source NLP Framework for Turkic Languages. Retrieved from http://zemberek.googlecode.com

Amasyali, F., Diri, B., & Turkoglu, F. (2006). Farklı Özellik Vektörleri ile Türkçe Dokümanların Yazarlarının Belirlenmesi. *Turkish Symposium on Artificial Intelligence and Neural Networks*, Turkey.

Amasyali, M. F., Balci, S., Varli, E.N., & Mete, E. (2012). A Comparison of Text Representation Methods for Turkish Text Classification. *EMO 2012.*

Amasyali, M. F., & Beken, A. (2009). Measurement of Turkish Word Semantic Similarity and Text Categorization Application. *SIU 2009,* Antalya.

Amasyali, M. F., Cetin, M., & Akbulut C. (2012). Metinlerin Anlamsal Uzayda Temsil Yöntemlerinin Sınıflandırma Performansına Etkileri. *ASYU 2012*.

Amasyali, M. F., Davletov, F., Torayew, A., & Ciftci, U. (2010). text2arff: Türkçe Metinler İçin Özellik Çıkarım Yazılımı. *SIU 2010*, Diyarbakır.

Amasyali, M.F., & Diri, B. (2006). Automatic Written Turkish Text Categorization in Terms of Author, Genre and Gender. *Proceedings of the 11th International Conference on Applications of Natural Language to Information Systems, LNCS,* 2006.

Amasyali, M.F., & Yildirim, T. (2004). Otomatik Haber Metinleri Sınıflandırma. *SIU2004.*

Cataltepe, Z., Turan, Y., & Kesgin, F. (2007). Turkish Document Classification Using Shorter Roots , *SIU 2007*, Eskisehir, Turkey.

Deng, S., & Peng, H. (2006). Document Classification Based on Support Vector Machine Using a Concept Vector Model. *The IEEE/WIC/ACM International Conference on Web Intelligence, WI 2006,* 473-476.

Guran, A., Akyokus, S., Guler, N., & Gurbuz, Z. (2009). Turkish Text Categorization Using N-gram Words, *INISTA 2009.*

Hall, M. A. & Smith, L. A. (1998). Practical Feature Subset Selection for Machine Learning. *Australian Computer Science Conference*. Springer. 181-191.

Hall, M. A. (1999). Correlation-based Feature Selection for Machine Learning. *Dept of Computer Science, University of Waikato.* Retrieved from http://www.cs.waikato.ac.nz/~mhall/thesis.pdf

Ishii, N., Murai, T., & Yamada, T. (2006). Text Classification by Combining Grouping, LSA and kNN. *Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and Proceedings of the 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse, IEEE,* 148-154.

Kazmierska, J. & Malicki, J. (2008). Application of the Naive Bayesian Classifier to Optimize Treatment Decisions. 211-216

Lee, C., Yang, H. & Ma, S. (2006). A Novel Multilingual Text Categorization System using Latent Semantic Indexing. *Proceedings of the 1st International Conference on Innovative Computing, Information and Control, IEEE, ICICIC 2006,* 503-506.

Lee, C., Yang, H., Chen, T., & Ma, S. (2006). A Comparative Study on Supervised and Unsupervised Learning Approaches for Multilingual Text Categorization. *Proceedings of the 1st International Conference on Innovative Computing, Information and Control, IEEE, ICICIC 2006,* 511-514.

Li, C. H., & Park S. C. (2007). Artificial Neural Network for Document Classification Using Latent Semantic Indexing. *International Symposium on Information Technology Convergence*, *ISITC 2007,* 17-21.

Li, C. H., & Park, S. C. (2007). An Efficient Document Categorization Model Based on LSA and BPNN. *Proceedings of the 6th International Conference on Advanced Language Processing and Web Information Technology, ALPIT, IEEE Computer Society,* 9-14.

Li, Y. H., & Jain, A.K. (1998). Classification of Text Documents. *Proceedings of the 14th IEEE International Conference on Pattern Recognition*, 1295-1297.

Liang, J. (2004). SVM Multi-classifier and Web Document Classification. *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics,* 1347-1351.

Magatti, D., Stella, F., & Faini, M. (2009). A Software System for Topic Extraction and Document Classification. *The IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, WI-IAT 2009*, 283-286.

Manning, C.D., Raghavan P. & Schtze H. (2008). Introduction to Information Retrieval. *Cambridge University Press*, New York, USA.

Mohamed, H.K. (2007). Automatic Documents Classification. *International Conference on Computer Engineering & Systems, IEEE, ICCES 2007,* 33-37.

Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. *Morgan Kaufmann Publishers*.

Ramasundaram, S., & Victor, S. P. (2013). Algorithms for Text Categorization: A Comparative Study. *World Applied Sciences Journal 22*, 1232-1240.

Sahami, M., Dumais, S., Heckerman, D. & Horvitz, E. (1998). A Bayesian Approach to Filtering Junk E-Mail. 98-105

Salton, G. (1968). Search and Retrieval Experiments in Real-time Information Retrieval. 1082-1093.

Schenker, A., Last, M., Bunke, H., & Kandel, A. (2003). Classification of Web documents using a graph model. *Proceedings of the 7th International Conference on Document Analysis and Recognition, IEEE Computer Society*, 240-244.

Shi, X., Zhao, Y., & Dong, X. (2008). Web Page Categorization Based on k-NN and SVM Hybrid Pattern Recognition Algorithm. *Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery, IEEE Computer Society, FSKD 2008*, 523-527.

Sun, J., Chen, Z., Zeng, H., Lu, Y., Shi, C., & Ma, W. (2004). Supervised Latent Semantic Indexing for Document Categorization. *Proceedings of the 4th IEEE International Conference on Data Mining, ICDM 2004,* 535-538.

Taghva, K., & Vergara, J. (2008). Feature Selection for Document Type Classification. *Proceedings of the 5th International Conference on Information Technology: New Generations, ITNG 2008,* 179-182.

Toraman, C., Can, F., & Kocberber, S. (2011). Developing a Text Categorization Template for Turkish News Portals. *2011 International Symposium on Innovations in Intelligent Systems and Applications (INISTA),* 379-383.

Torunoglu, D., Cakırman, E., Ganiz, M.C., Akyokus, S., & Gurbuz, M.Z. (2011). Analysis of Preprocessing Methods on Classification of Turkish Texts. *International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 112-117.

Tufekci, P., Uzun, E., & Sevinc, B. (2012). Text Classification of Web Based News Articles by Using Turkish Grammatical Features, *IEEE 2012.*

Uzundere, E., Dedja, E., Diri, B. & Amasyalı, M.F. (2008). Türkçe Haber Metinleri için Otomatik Özetleme. *ASYU 2008*, Isparta.

Wang, Q., Garrity, G. M., Tiedje, J. M., & Cole, J. R. (2007). Naive Bayesian Classifier for Rapid Assignment of Rrna Sequences into the New Bacterial Taxonomy. 5261-5267.

Wang, Y., Hodges, Y., & Tang, B. (2003). Classification of Web Documents using a Naive Bayes Method. *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, 560-564.

Wang, Z., Liu, Y., & Sun, X. (2008). An Efficient Web Document Classification Algorithm Based on LPP and SVM. *Chinese Conference on Pattern Recognition, IEEE, CCPR*, 1-4.

Witten, I. H. & Frenk, E. (2005). Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufman, San Francisco.

Yildiz, H.K., Usta, N., Genctav, M., Diri, B., & Amasyali, M.F. (2007) A New Feature Extraction Method for Text Classification. *SIU 2007*, Eskişehir, Türkiye.

Zhang, Y., Fan, B., & Xiao, L. B. (2008). Web Page Classification Based on a Least Square Support Vector Machine with Latent Semantic Analysis. *Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery, IEEE, FSKD 2008*, 528-532.

APPENDIX A

STOP WORD LIST WITH 356 WORDS

a
acaba
altı
altmış
ama
ancak
arada
artık
asla
aslında
ayrıca
az
b
bana
bazen
bazı
bazıları
bazısı
belki
ben
benden
beni
benim
beri
beş
bile
bin
bir
birçoğu
birçok
birçokları
biri
birisi
birkaç
birkaçı
birkez
birşey
birşeyi
biz
bizden
bize
bizi
bizim
böyle
böylece
bu
buna
bunda
bundan
bunlar
bunları
bunların
bunu
bunun
burada
bütün
c
ç
çoğu
çoğuna
çoğunu
çok
çünkü
d
da
daha
dahi
de
defa
değil
demek
diğer
diğeri
diğerleri
diye
doksan
dokuz
dolayı
dolayısıyla
dört
e
edecek
eden
ederek

edilecek
ediliyor
edilmesi
ediyor
eğer
elbette
elli
en
et
etmek
etmesi
etti
ettiği
ettiğini
ettik
ettiler
ettim
ettin
ettiniz
eyle
eyledi
eyledik
eylediler
eyledim
eyledin
eylediniz
eylemek
f
fakat
falan
felan
filan
g
gene
gibi
göre
ğ
h
hâlâ
halen
hangi
hangisi
hani
hatta
hem
henüz
hep
hepsi
hepsine
hepsini
her
her biri
herhangi
herkes
herkese
herkesi
herkesin
hiç
hiç kimse
hiçbir
hiçbiri
hiçbirine
hiçbirini
ı
i
için
içinde
iki
ile
ilgili
ise
işte
itibaren
itibariyle
j
k
kaç
kadar
karşın
katrilyon
kendi
kendilerine
kendine
kendini
kendisi
kendisine
kendisini
kez
kıl
kıldı
kıldık
kıldılar
kıldım
kıldın
kıldınız
kılmak

kırk
ki
kim
kimden
kime
kimi
kimin
kimisi
kimse
l
la
lakin
le
li
lik
lık
lu
m
madem
mı
mi
milyar
milyon
mu

mü
n
nan
nasıl
ne
ne kadar
ne zaman
neden
nedenle
nedir
nen
nerde
nerede
nereden
nereye
nesi
neyse
nın
niçin
niye
nun
o
ol
olan

olarak
oldu
olduğu
olduğunu
olduk
olduklarını
oldular
oldum
oldun
oldunuz
olmadı
olmadığı
olmak
olması
olmayan
olmaz
olsa
olsun
olup
olur
olursa
oluyor
on
ona

ondan
onlar
onlara
onlardan
onları
onların
onu
onun
orada
otuz
oysa
oysaki
ö
öbürü
ön
önce
ötürü
öyle
p
pek
r
rağmen
s
sadece

sana
sanki
sekiz
seksen
sen
senden
seni
senin
siz
sizden
size
sizi
sizin
son
sonra
ş
şayet
şey
şeyden
şeye
şeyi
şeyler
şimdi
şöyle
şu
şuna
şunda
şundan
şunlar
şunları
şunu
şunun
t
tabi
tamam
tarafından
trilyon
tüm
tümü
u
ü
üç
üzere
v
var
vardı
ve
veya
veyahut
y
ya
ya da
yani
yap
yapacak
yapılan
yapılması
yapıyor
yapmak
yapmak
yaptı
yaptığı
yaptığını
yaptık
yaptıkları
yaptılar
yaptım
yaptın
yaptınız
yedi
yerine
yine
yirmi
yoksa
yüz
z
zaten
zi

APPENDIX B

BW – NB CLASSIFICATION RESULTS

Table B.1. BW and NB classification results

		Accuracy	Precision	Recall	F-Measure	Correct	Incorrect
Original	**No feature selection**	68,23%	0,709	0,682	0,691	39271	18282
	IG	68,25%	0,709	0,682	0,692	39279	18274
	X^2	68,25%	0,709	0,682	0,692	39279	18274
	CBFS	65,43%	0,677	0,654	0,662	37657	19896
	Bi-gram	47,45%	0,621	0,474	0,512	27307	30246
	Bi-gram + IG	47,45%	0,621	0,474	0,512	27308	30245
	Bi-gram + X^2	47,45%	0,621	0,474	0,512	27308	30245
	Bi-gram + CBFS	50,34%	0,509	0,503	0,496	28973	28580
	Tri-gram	54,73%	0,668	0,547	0,572	31500	26053
	Tri-gram + IG	54,73%	0,668	0,547	0,572	31499	26054
	Tri-gram + X^2	54,73%	0,668	0,547	0,572	31499	26054
	Tri-gram + CBFS	62,87%	0,645	0,629	0,633	36186	21367
Stem	**No feature selection**	68,80%	0,716	0,688	0,698	39595	17958
	IG	68,79%	0,716	0,688	0,698	39591	17962
	X^2	68,79%	0,716	0,688	0,698	39591	17962
	CBFS	67,23%	0,687	0,672	0,677	38692	18861
	Bi-gram	48,54%	0,625	0,485	0,520	27938	29615
	Bi-gram + IG	48,55%	0,625	0,486	0,520	27943	29610
	Bi-gram + X^2	48,55%	0,625	0,486	0,520	27943	29610
	Bi-gram + CBFS	51,45%	0,517	0,515	0,507	29613	27940
	Tri-gram	57,68%	0,677	0,577	0,599	33197	24356
	Tri-gram + IG	57,69%	0,677	0,577	0,599	33203	24350
	Tri-gram + X^2	57,69%	0,677	0,577	0,599	33203	24350
	Tri-gram + CBFS	63,57%	0,657	0,636	0,640	36584	20969

APPENDIX C

BW – C4.5 CLASSIFICATION RESULTS

Table C.1. BW and C4.5 Classification Results

		Accuracy	Precision	Recall	F-Measure	Correct	Incorrect
Original	**No feature selection**	92,40%	0,924	0,924	0,924	53177	4376
	IG	92,52%	0,925	0,925	0,925	53249	4304
	X^2	92,44%	0,924	0,924	0,924	53202	4351
	CBFS	71,50%	0,714	0,715	0,710	41148	16405
	Bi-gram	88,90%	0,889	0,889	0,888	51164	6389
	Bi-gram + IG	88,99%	0,890	0,890	0,890	51216	6337
	Bi-gram + X^2	89,04%	0,891	0,890	0,890	51247	6306
	Bi-gram + CBFS	61,81%	0,620	0,618	0,612	35576	21977
	Tri-gram	92,50%	0,925	0,925	0,925	53235	4318
	Tri-gram + IG	92,46%	0,925	0,925	0,924	53212	4341
	Tri-gram + X^2	92,47%	0,925	0,925	0,924	53221	4332
	Tri-gram + CBFS	72,24%	0,720	0,722	0,718	41576	15977
Stem	**No feature selection**	92,47%	0,925	0,925	0,924	53220	4333
	IG	92,50%	0,925	0,925	0,925	53235	4318
	X^2	92,48%	0,925	0,925	0,924	53223	4330
	CBFS	75,53%	0,754	0,755	0,752	43469	14084
	Bi-gram	89,05%	0,891	0,890	0,890	51249	6304
	Bi-gram + IG	89,12%	0,892	0,891	0,891	51291	6262
	Bi-gram + X^2	89,08%	0,891	0,891	0,890	51271	6282
	Bi-gram + CBFS	63,20%	0,630	0,632	0,626	36374	21179
	Tri-gram	92,38%	0,924	0,924	0,924	53168	4385
	Tri-gram + IG	92,29%	0,923	0,923	0,923	53115	4438
	Tri-gram + X^2	92,31%	0,923	0,923	0,923	53127	4426
	Tri-gram + CBFS	78,63%	0,785	0,786	0,784	45255	12298

APPENDIX D

BW – SVM CLASSIFICATION RESULTS

Table D.1. BW and SVM Classification Results

		Accuracy	Precision	Recall	F-Measure	Correct	Incorrect
Original	**No feature selection**	93,61%	0,936	0,936	0,936	53874	3679
	IG	93,14%	0,931	0,931	0,931	53606	3947
	X^2	93,14%	0,931	0,931	0,931	53606	3947
	CBFS	67,32%	0,677	0,673	0,670	38747	18806
	Bi-gram	78,47%	0,784	0,785	0,784	45161	12392
	Bi-gram + IG	78,20%	0,781	0,782	0,781	45004	12549
	Bi-gram + X^2	78,21%	0,781	0,782	0,781	45011	12542
	Bi-gram + CBFS	53,46%	0,533	0,535	0,527	30766	26787
	Tri-gram	89,81%	0,897	0,898	0,898	51691	5862
	Tri-gram + IG	89,81%	0,897	0,898	0,898	51691	5862
	Tri-gram + X^2	89,82%	0,898	0,898	0,898	51693	5860
	Tri-gram + CBFS	66,08%	0,660	0,661	0,656	38033	19520
Stem	**No feature selection**	92,35%	0,923	0,923	0,923	53149	4404
	IG	92,34%	0,923	0,923	0,923	53147	4406
	X^2	92,34%	0,923	0,923	0,923	53146	4407
	CBFS	70,37%	0,707	0,704	0,700	40498	17055
	Bi-gram	78,58%	0,785	0,786	0,785	45225	12328
	Bi-gram + IG	78,38%	0,783	0,784	0,783	45109	12444
	Bi-gram + X^2	78,37%	0,783	0,784	0,783	45104	12449
	Bi-gram + CBFS	54,84%	0,541	0,548	0,537	31560	25993
	Tri-gram	90,77%	0,907	0,908	0,907	52242	5311
	Tri-gram + IG	90,77%	0,907	0,908	0,907	52241	5312
	Tri-gram + X^2	90,77%	0,907	0,908	0,907	52242	5311
	Tri-gram + CBFS	68,61%	0,685	0,686	0,682	39486	18067

APPENDIX E

TF – NB CLASSIFICATION RESULTS

Table E.1. TF and NB Classification Results

		Accuracy	Precision	Recall	F-Measure	Correct	Incorrect
Original	**No feature selection**	70,78%	0,731	0,708	0,716	40736	16817
	IG	72,41%	0,737	0,724	0,728	41673	15880
	X^2	72,30%	0,736	0,723	0,727	41611	15942
	CBFS	64,10%	0,689	0,641	0,643	36890	20663
	Bi-gram	47,23%	0,564	0,472	0,495	27181	30372
	Bi-gram + IG	47,63%	0,566	0,476	0,499	27410	30143
	Bi-gram + X^2	47,62%	0,566	0,476	0,498	27406	30147
	Bi-gram + CBFS	49,51%	0,594	0,495	0,499	28495	29058
	Tri-gram	58,98%	0,659	0,590	0,605	33942	23611
	Tri-gram + IG	59,19%	0,660	0,592	0,607	34064	23489
	Tri-gram + X^2	59,16%	0,660	0,592	0,606	34047	23506
	Tri-gram + CBFS	60,35%	0,660	0,603	0,606	34733	22820
Stem	**No feature selection**	71,97%	0,736	0,720	0,726	41419	16134
	IG	73,01%	0,740	0,730	0,734	42022	15531
	X^2	72,95%	0,740	0,730	0,733	41986	15567
	CBFS	62,30%	0,682	0,623	0,629	35856	21697
	Bi-gram	47,43%	0,563	0,474	0,497	27300	30253
	Bi-gram + IG	47,83%	0,565	0,478	0,500	27530	30023
	Bi-gram + X^2	47,83%	0,565	0,478	0,500	27530	30023
	Bi-gram + CBFS	50,31%	0,598	0,503	0,507	28956	28597
	Tri-gram	61,61%	0,669	0,616	0,631	35458	22095
	Tri-gram + IG	61,86%	0,670	0,619	0,632	35604	21949
	Tri-gram + X^2	61,82%	0,670	0,618	0,632	35581	21972
	Tri-gram + CBFS	62,33%	0,669	0,623	0,626	35872	21681

APPENDIX F

TF – C4.5 CLASSIFICATION RESULTS

Table F.1. TF and C4.5 Classification Results

		Accuracy	Precision	Recall	F-Measure	Correct	Incorrect
Original	**No feature selection**	92,72%	0,927	0,927	0,927	53363	4190
	IG	92,84%	0,928	0,928	0,928	53435	4118
	X^2	92,88%	0,929	0,929	0,928	53457	4096
	CBFS	81,05%	0,810	0,811	0,809	46647	10906
	Bi-gram	91,19%	0,912	0,912	0,912	52485	5068
	Bi-gram + IG	91,31%	0,913	0,913	0,913	52550	5003
	Bi-gram + X^2	91,25%	0,913	0,912	0,912	52516	5037
	Bi-gram + CBFS	78,89%	0,788	0,789	0,788	45404	12149
	Tri-gram	93,52%	0,935	0,935	0,935	53823	3730
	Tri-gram + IG	93,49%	0,935	0,935	0,935	53806	3747
	Tri-gram + X^2	93,54%	0,936	0,935	0,935	53835	3718
	Tri-gram + CBFS	85,11%	0,850	0,851	0,850	48982	8571
Stem	**No feature selection**	92,94%	0,929	0,929	0,929	53488	4065
	IG	92,92%	0,929	0,929	0,929	53478	4075
	X^2	92,90%	0,929	0,929	0,929	53469	4084
	CBFS	81,40%	0,813	0,814	0,812	46847	10706
	Bi-gram	91,19%	0,912	0,912	0,912	52484	5069
	Bi-gram + IG	91,34%	0,914	0,913	0,913	52571	4982
	Bi-gram + X^2	91,35%	0,914	0,914	0,913	52577	4976
	Bi-gram + CBFS	78,48%	0,783	0,785	0,783	45165	12388
	Tri-gram	93,39%	0,934	0,934	0,934	53747	3806
	Tri-gram + IG	93,40%	0,934	0,934	0,934	53756	3797
	Tri-gram + X^2	93,53%	0,935	0,935	0,935	53832	3721
	Tri-gram + CBFS	86,40%	0,863	0,864	0,863	49727	7826

APPENDIX G

TF – SVM CLASSIFICATION RESULTS

Table G.1. TF and SVM Classification Results

		Accuracy	Precision	Recall	F-Measure	Correct	Incorrect
Original	No feature selection	90,14%	0,901	0,901	0,901	51878	5675
	IG	89,76%	0,897	0,898	0,897	51662	5891
	X^2	89,77%	0,897	0,898	0,897	51667	5886
	CBFS	71,42%	0,721	0,714	0,712	41103	16450
	Bi-gram	80,41%	0,803	0,804	0,804	46276	11277
	Bi-gram + IG	80,19%	0,801	0,802	0,801	46151	11402
	Bi-gram + X^2	80,19%	0,801	0,802	0,801	46153	11400
	Bi-gram + CBFS	60,14%	0,606	0,601	0,593	34611	22942
	Tri-gram	87,98%	0,879	0,880	0,879	50634	6919
	Tri-gram + IG	87,97%	0,879	0,880	0,879	50632	6921
	Tri-gram + X^2	87,97%	0,879	0,880	0,879	50630	6923
	Tri-gram + CBFS	71,21%	0,713	0,712	0,708	40981	16572
Stem	No feature selection	89,14%	0,891	0,891	0,891	51301	6252
	IG	89,04%	0,890	0,890	0,890	51248	6305
	X^2	89,04%	0,890	0,890	0,890	51247	6306
	CBFS	72,26%	0,727	0,723	0,719	41590	15963
	Bi-gram	80,36%	0,802	0,804	0,802	46250	11303
	Bi-gram + IG	80,22%	0,801	0,802	0,801	46170	11383
	Bi-gram + X^2	80,22%	0,801	0,802	0,801	46169	11384
	Bi-gram + CBFS	59,87%	0,605	0,599	0,591	34458	23095
	Tri-gram	88,34%	0,882	0,883	0,883	50844	6709
	Tri-gram + IG	88,35%	0,882	0,883	0,883	50847	6706
	Tri-gram + X^2	88,35%	0,882	0,883	0,883	50848	6705
	Tri-gram + CBFS	73,10%	0,732	0,731	0,728	42072	15481

APPENDIX H

TF-IDF – NB CLASSIFICATION RESULTS

Table H.1. TF-IDF and NB Classification Results

		Accuracy	Precision	Recall	F-Measure	Correct	Incorrect
Original	**No feature selection**	70,80%	0,731	0,708	0,716	40748	16805
	IG	72,40%	0,737	0,724	0,728	41668	15885
	X^2	72,30%	0,736	0,723	0,727	41612	15941
	CBFS	64,09%	0,689	0,641	0,643	36887	20666
	Bi-gram	47,25%	0,564	0,473	0,495	27196	30357
	Bi-gram + IG	47,64%	0,566	0,476	0,499	27416	30137
	Bi-gram + X^2	47,63%	0,566	0,476	0,499	27412	30141
	Bi-gram + CBFS	49,48%	0,594	0,495	0,498	28479	29074
	Tri-gram	58,96%	0,659	0,590	0,605	33936	23617
	Tri-gram + IG	59,18%	0,660	0,592	0,607	34059	23494
	Tri-gram + X^2	59,17%	0,660	0,592	0,607	34053	23500
	Tri-gram + CBFS	60,35%	0,660	0,603	0,606	34733	22820
Stem	**No feature selection**	72,00%	0,736	0,720	0,726	41441	16112
	IG	73,05%	0,740	0,730	0,734	42041	15512
	X^2	72,99%	0,740	0,730	0,734	42010	15543
	CBFS	62,30%	0,682	0,623	0,629	35856	21697
	Bi-gram	47,49%	0,564	0,475	0,497	27332	30221
	Bi-gram + IG	47,87%	0,565	0,479	0,501	27551	30002
	Bi-gram + X^2	47,88%	0,565	0,479	0,501	27554	29999
	Bi-gram + CBFS	50,31%	0,599	0,503	0,508	28954	28599
	Tri-gram	61,66%	0,669	0,617	0,631	35487	22066
	Tri-gram + IG	61,90%	0,670	0,619	0,633	35627	21926
	Tri-gram + X^2	61,83%	0,670	0,618	0,632	35584	21969
	Tri-gram + CBFS	62,33%	0,669	0,623	0,626	35872	21681

APPENDIX I

TF-IDF – C4.5 CLASSIFICATION RESULTS

Table I.1. TF-IDF and C4.5 Classification Results

		Accuracy	Precision	Recall	F-Measure	Correct	Incorrect
Original	**No feature selection**	92,72%	0,927	0,927	0,927	53363	4190
	IG	92,84%	0,928	0,928	0,928	53435	4118
	X^2	92,88%	0,929	0,929	0,928	53457	4096
	CBFS	81,05%	0,810	0,811	0,809	46647	10906
	Bi-gram	91,19%	0,912	0,912	0,912	52485	5068
	Bi-gram + IG	91,31%	0,913	0,913	0,913	52550	5003
	Bi-gram + X^2	91,25%	0,913	0,912	0,912	52516	5037
	Bi-gram + CBFS	78,89%	0,788	0,789	0,788	45404	12149
	Tri-gram	93,52%	0,935	0,935	0,935	53823	3730
	Tri-gram + IG	93,49%	0,935	0,935	0,935	53806	3747
	Tri-gram + X^2	93,54%	0,936	0,935	0,935	53835	3718
	Tri-gram + CBFS	85,11%	0,850	0,851	0,850	48982	8571
Stem	**No feature selection**	92,94%	0,929	0,929	0,929	53488	4065
	IG	92,92%	0,929	0,929	0,929	53478	4075
	X^2	92,90%	0,929	0,929	0,929	53469	4084
	CBFS	81,40%	0,813	0,814	0,812	46847	10706
	Bi-gram	91,19%	0,912	0,912	0,912	52484	5069
	Bi-gram + IG	91,34%	0,914	0,913	0,913	52571	4982
	Bi-gram + X^2	91,35%	0,914	0,914	0,913	52577	4976
	Bi-gram + CBFS	78,48%	0,783	0,785	0,783	45165	12388
	Tri-gram	93,39%	0,934	0,934	0,934	53747	3806
	Tri-gram + IG	93,40%	0,934	0,934	0,934	53756	3797
	Tri-gram + X^2	93,53%	0,935	0,935	0,935	53832	3721
	Tri-gram + CBFS	86,40%	0,863	0,864	0,863	49727	7826

APPENDIX J

TF-IDF – SVM CLASSIFICATION RESULTS

Table J.1. TF-IDF and SVM Classification Results

		Accuracy	Precision	Recall	F-Measure	Correct	Incorrect
Original	**No feature selection**	90,15%	0,901	0,901	0,901	51882	5671
	IG	89,77%	0,897	0,898	0,897	51664	5889
	X^2	89,77%	0,897	0,898	0,897	51666	5887
	CBFS	71,43%	0,721	0,714	0,712	41110	16443
	Bi-gram	80,40%	0,803	0,804	0,803	46270	11283
	Bi-gram + IG	80,20%	0,801	0,802	0,801	46159	11394
	Bi-gram + X^2	80,19%	0,801	0,802	0,801	46153	11400
	Bi-gram + CBFS	60,14%	0,606	0,601	0,593	34612	22941
	Tri-gram	87,98%	0,879	0,880	0,879	50635	6918
	Tri-gram + IG	87,98%	0,879	0,880	0,879	50633	6920
	Tri-gram + X^2	87,98%	0,879	0,880	0,879	50634	6919
	Tri-gram + CBFS	71,21%	0,713	0,712	0,708	40981	16572
Stem	**No feature selection**	89,13%	0,890	0,891	0,891	51297	6256
	IG	89,04%	0,890	0,890	0,890	51246	6307
	X^2	89,04%	0,890	0,890	0,890	51245	6308
	CBFS	72,26%	0,727	0,723	0,719	41585	15968
	Bi-gram	80,37%	0,802	0,804	0,803	46257	11296
	Bi-gram + IG	80,22%	0,801	0,802	0,801	46168	11385
	Bi-gram + X^2	80,20%	0,801	0,802	0,801	46158	11395
	Bi-gram + CBFS	59,90%	0,606	0,599	0,592	34473	23080
	Tri-gram	88,35%	0,882	0,883	0,883	50846	6707
	Tri-gram + IG	88,34%	0,882	0,883	0,882	50840	6713
	Tri-gram + X^2	88,34%	0,882	0,883	0,883	50844	6709
	Tri-gram + CBFS	73,09%	0,732	0,731	0,728	42067	15486