APPLICATION OF K-NN AND FPTC BASED TEXT CATEGORIZATION ALGORITHMS TO TURKISH NEWS REPORTS

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER

ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

by

Ufuk Ilhan

February, 2001

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

Assoc. Prof. Dr. Halil Altay Güvenir (Advisor)

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

Assoc. Prof. Dr. Cevdet Aykanat

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

Asst. Prof. Dr. İlyas Çiçekli

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Science

ABSTRACT

APPLICATION OF k-NN and FPTC BASED TEXT CATEGORIZATION ALGORITHMS TO TURKISH NEWS REPORTS

Ufuk Ilhan
M.S. in Computer Engineering
Supervisor: Assoc. Prof. Halil Altay Güvenir
February, 2001

New technological developments, such as easy access to Internet, optical character readers, high-speed networks and inexpensive massive storage facilities, have resulted in a dramatic increase in the availability of on-line text-newspaper articles, incoming (electronic) mail, technical reports, etc. The enormous growth of on-line information has led to a comparable growth in the need for methods that help users organize such information. Text Categorization may be the remedy of increased need for advanced techniques. Text Categorization is the classification of units of natural language texts with respect to a set of pre-existing categories. Categorization of documents is challenging, as the number of discriminating words can be very large. This thesis presents compilation of a Turkish dataset, called Anadolu Agency Newsgroup in order to study in Text Categorization. Turkish is an agglutinative languages in which words contain no direct indication where the morpheme boundaries are, furthermore, morphemes take a shape dependent on the morphological and phonological context. In Turkish, the process of adding one suffix to another can result in a relatively long word, furthermore, a single Turkish word can give rise to a very large number of variants. Due to this complex morphological structure, Turkish requires text processing techniques different than English and similar languages. Therefore, besides converting all words to lower case and removing punctuation marks, some preliminary work is required such as stemming, removal of stopwords and formation of a keyword list.

This thesis also presents the evaluation and comparison of the well-known k-NN classification algorithm and a variant of the k-NN, called Feature Projection Text Categorization (FPTC) algorithm. The k-NN classifier is an instance based learning method. It computes the similarity between the test instance and training instance, and considering the k top-ranking nearest instances to predict the categories of the input, finds out the category that is most similar. FPTC algorithm is based on the idea of representing training instances as their projections on each feature dimension. If the value of a training instance is missing for a feature, that instance is not stored on that feature. Experiments show that the FPTC algorithm achieves comparable accuracy with the k-NN algorithm, furthermore, the time efficiency of FPTC outperforms the k-NN significantly.

Keywords: text categorization, classification, feature projections, stemming, wild card matching, stopword.

ÖZET

k-NN ve FPTC TABANLI METİN KATEGORİZASYON ALGORİTMALARININ TÜRKÇE HABERLERE UYGULAMASI

Ufuk Ilhan
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Doç. Dr. Halil Altay Güvenir
Şubat, 2001

İnternet ulaşım kolaylığı, optik okuyucular, yüksek hızlı ağlar ve pahalı olmayan yüksek miktardaki bilgi depolama imkanlarındaki teknolojik gelişmeler, on-line metin ve makalelerine, elektronik posta ve teknik raporlara erişim kolaylığıyla büyük bir artışa neden oldu. On-line bilgi erişimindeki, bu inanılmaz artış, kullanıcı ların bilgileri organize etme ihtiyacını yarattı.

Metin sınıflandırması (Text Categorization), gelişen tekniklerin ihtiyaçlarına bir çare olabilir. Metin sınıflandırması, önceden belirlenmiş kategorilere göre, doğal dil metinlerinin sınıflandırılmasıdır. Bu tezde, metin sınıflandırması üzerinde çalışmak için Anadolu Ajansı adlı Türkçe bir veri kümesinin derlenmesi sunulmuştur. Türkçe gibi bitişken dillerde kelimeler, en küçük anlamlı parçasının sınırlarına dair bir belirti göstermez, üstelik, bu parçalar, morfolojik ve fonolojik şartlara bağlı olarak şekil alırlar. Türkçe'de, bir kelimenin son ekine bir tane daha ekleyerek, nispeten uzun kelimeler elde edilebilir, üstelik, sadece bir tek Türkçe kelimeden çok miktarda değişik anlamlı kelimeler oluşturulabilir. Bu karmaşık morfolojik yapı yüzünden, Türkçe, İngilizce ve benzer dillerden daha farklı metin özel işlem teknikleri gerektirir. Bu nedenle, bütün kelimelerin küçük harfe çevrilmesi ve noktalama işaretlerinin atılması dışında; gövdeleme, gereksiz kelimelerin atılması ve anahtar kelime listesinin oluşturulması gibi, bazı önhazırlıklar yapılması gereklidir.

Bu tezde, ayrıca, literatürde yaygın olarak bilinen k en yakın komşu sınıflandırma algoritması (k-NN) ile k-NN'in bir değişiği olan FPTC algoritmasının Türkçe veri kümesi üzerinde değerlemesi ve karşılaştırılması da sunulmuştur. k-NN, bir örnek tabanlı öğrenme metodudur. k-NN, tahmin ve test örnekleri arasındaki benzerliği hesaplar ve girdi kategorilerini tahmin etmek için k adet üst sıranın en yakın örneklerini düşünerek, en benzer kategorileri bulur. FPTC algoritması ise, tahmin örneklerinin izdüşümlerinin, herbir öznitelik boyutunda ifade edilmesi fikri esasına dayalıdır. Eğer, bir tahmin örneğinin değeri, bir öznitelik için belli değilse, tahmin örneği, öznitelik üzerinde ifade edilmez. Yapılan değerlemeler sonucu, FPTC algoritması, k-NN'le karşılaştırılabilir bir doğruluk oranını başarmıştır, ayrıca, zaman verimliliği açısından, k-NN algoritmasına belirgin bir üstünlük sağlamıştır.

Eşime, Anneme, Babama ve Kardeşime

I would like to express my gratitude to Dr. H. Altay Güvenir, from whom I have learned a lot, due to his supervision, suggestions, and support during this research.

I am also indebted to Dr. Cevdet Aykanat and Dr. İlyas Çiçekli for showing keen interest to the subject matter and accepting to read and review this thesis.

Contents

List of Figures

List of Tables

List of Symbols and Abbreviations

B	: Basis Function
β	: Parameter set
CART	: Classification and Regression Trees
d	: Distance function
D	: Training set
DART	: Regression Tree Induction Algorithm
DMSK	: Data Miner Software Kit
DNF	: Disjunctive Normal Form
f	: Approximated function
I	: Impurity measure
i	: Instance
IBL	: Instance-Based Learning
K	: Kernel Function
k	: Number of neighbor instances
KMEANS	: Partitioning clustering algorithm
KNN	: K Nearest Neighbor
KDD	: Knowledge Discovery in Databases
L	: Loss function
log	: Logarithm in base 2
m	: Number of predictor features
MAD	: Mean Absolute Distance
MARS	: Multivariate Adaptive Regression Splines
M5	: Regression tree induction algorithm
n	: Number of training instances
p	: Number of parameters or features
$\mathbf{x_q}$	: Query instance
R	: Region
R_k	: Rule set
RE	: Relative Error
RETIS	: Regression tree induction algorithm
RSBF	: Regression by Selecting Best Features
RSBFP	: Regression by Selecting Best Feature Projections

RULE	: Rule-based regression algorithm
r	: Rule
t	: A test example
T	: Number of test instances
$\mathbf{X}$	: Instance matrix
$\mathbf{x}$	: Instance vector
$\mathbf{x}_i$	: Value vector of i^{th} instance
$\mathbf{y}$	: Target vector
$\bar{y}$	: Estimated target

Chapter 1

Introduction

New technological developments, such as easy access to Internet, optical character readers, high-speed networks and inexpensive massive storage facilities, have resulted in a dramatic increase in the availability of on-line text-newspaper articles, incoming (electronic) mail, technical reports, etc. The enormous growth of on-line information has led to a comparable growth in the need for methods that help users organize such information.

Text Categorization may be the remedy of increased need for advanced techniques. Text Categorization is the classification of units of natural language texts with respect to a set of pre-existing categories. Reducing an infinite set of possible natural language inputs to a small set of categories is a central strategy in computational systems that process textual information.

Text Categorization has become important in two aspects. From the Information Retrieval (IR) point of view, information processing needs have increased with the rapid growth of textual information sources, such as Internet. Text Categorization can be used to support IR or to perform information extraction, document filtering and routing to topic-specific processing mechanisms. From the Machine Learning (ML) point of view, recent research has been concerned with scaling up (e.g. data mining). Text Categorization is a domain where large data sets are available and which provides an application field to ML. Indeed, manual categorization is known to be an expensive

and time-consuming task which results are dependent on variations in experts' judgements [24].

There has been an recent outbreak of application and usage of Text Categorization, especially not only assigning subject categories to documents in support of text retrieval and library organization, but also aiding the human assignment of such categories. However, while routing messages, news stories or other continuous streams of texts to interested recipients; Text Categorization is used. As a component in natural language processing systems, to filter out non-relevant texts and parts of texts, to route texts to category-specific processing mechanisms or to extract limited forms of information and also as an aid in lexical anlysis tasks, such as word sense disambiguation, are examples of usage areas of Text Categorization.

There are two basic selection steps while studying in Text Categorization. The first one is to select a categorization algorithm to evaluate the performance, the other is to select a sample data collection on which the algorithm is applied. In the following section, the dataset used in this thesis is introduced.

1.1 Anadolu Agency Dataset

Ideally, all researchers would like to use a common data collection and compare performance measures to evaluate their systems. The sample dataset is important for both the effectiveness and the efficiency of statistical text categorization. That is, researchers would like a training set which contains sufficient information for example-based learning of categorization, but is not too large for efficient computation. The latter is particularly important for solving large categorization problems in practical databases [39].

Nearly all researchers have been concerned with English or with languages morphologically similar to English. In such languages, words contain only a small number of affixes, or none at all, almost all of parsing models for them consider recognizing those affixes as being trivial, and thus do not make morphological analyses. This feature allows easy stemming of the words to

find their root words. On the other hand, agglutinative languages as Turkish, words contain no direct indication where the morpheme boundaries are, and furthermore morphemes take a shape dependent on the morphological and phonological context [26]. The establishment of independence for the new Turkic republics necessitates creating their own industry [3]. It is doubtless that there is a serious problem in *Text Categorization* evaluation because of the lack of standard Turkish dataset regarding to meet these requirements.

In this thesis, we will cocern with *Anadolu Agency News Dataset* to meet the requirements. The dataset consists of nearly 200 000 unprocessed Turkish news documents (Fig 1.1), but only 2000 of them is processed for the present (Fig 1.2). Each news report contains a categorized number body, a headline text and news text body. The headlines are an average of 12 words long. The average length of a document body is 96 words. On average, 7 categories are assigned to each document. There are many "noisy" data which makes the categorization difficult to learn for a categorizer. The original A.A. (Anadolu Agency) dataset is unprocessed that is the categories were manually assigned to subjects using 78 subject categories. Each category label is represented by a number defined a subject. Word boundaries were defined by whitespace. Some preliminary work is required besides converting all words to lower case and removing punctuation marks because of the characteristics of Turkish language. The preprocessing work is described in more detail in Chapter 4.

1.1.1 The Characteristics of Turkish Language

Turkish is a member of the south-western or Oghuz group of the Turkic languages, which also includes Turkmen, Azerbaijani, Ghasghai and Gagaus. The Turkish language uses a form of Latin alphabet consisting of twenty-nine letters, of which eight are vowels and twenty-one are consonants. Unlike the main Indo-European languages, such as French, English and German, Turkish is an example of an agglutinative language, where words are formed by affixing morphemes to a root in order to extend its meaning or to create other classes of words. In Turkish, the process of adding one suffix to another can result in a relatively long word, which often contains an amount of semantic

ANKARA'DA OKULLARA KAR TATİLİ...

ANKARA (A.A) - Ankara'da kar yağışı nedeniyle okulların bugün tatil edildiği bildirildi.

Ankara Valiliği'nden yapılan açıklamada, Ankara'da iki gündür etkili olan kar yağışı sebebiyle merkez ilçelerinde bulunan ilköğretim, lise ve dengi okulların bugün tatil edildiği bildirildi.
(CÜN-SRP)
07:25 04/01/00

TRAFİK KAZASI: 1 ÖLÜ...

ADANA (A.A) - Adana'da meydana gelen trafik kazasında bir kişi öldü.

Alınan bilgiye göre, sürücünün kimliği ve plakası belirlenemeyen bir araç, Ziyapaşa Bulvarı'nda yolun karşısına geçmek isteyen Şükrü Bulan'a (80) çarparak, ölümüne neden oldu.

Kaçan araç sürücüsünün yakalanmasına çalşıldığı bildirildi.
(DA-CÜN-SRP)
07:51 04/01/00

ARTÇI SARSINTILAR SÜRÜYOR...

ISTANBUL (A.A) - Düzce'de 12 Kasım 1999'da meydana gelen depremin artçı sarsıntıları sürüyor.
Boğaziçi Üniversitesi Kandilli Rasathanesi ve Deprem Araştırma Enstitüsü'nden verilen bilgiye göre, bugün saat 02.28'de Düzce'de 3.2 büyüklüğünde bir artçı sarsıntı kaydedildi.
(MER-CÜN-İDA)
08:16 03/01/00

Figure 1.1: The Original Unprocessed News Report

information equivalent to a whole English phrase, clause or sentence. Due to this complex morphological structure, a single Turkish word can give rise to a very large number of variants. The experiments [8] show that the use of a stopword list and a stemming procedure can bring about substantial reductions in the numbers of word variants encountered in searches of Turkish text datasets; moreover, stemming appears to be superior. However, stemming in an agglutinative language is quite complex.

As a preliminary work in the thesis, we have also decided which words are in stemming word list, and then which words are in stopword list or keyword list.

1 7 78 | ankara okullara kar tatili | ankara kar yağışı okulların tatil edildiği ankara valiliği açıklamada ankara gündür etkili kar yağışı merkez ilçelerinde ilköğretim lise dengi okulların tatil edildiği
1 19 23 | trafik kazası ölü | adana meydana gelen trafik kazasında kişi öldü sürücüsünün kimliği plakası belirlenemeyen bir araç ziyapaşa bulvarı yolun karşısına geçmek isteyen şükrü bulan çarparak ölümüne neden oldu kaçan araç sürücüsünün yakalanmasına çalşıldığı bildirildi.
1 7 69 71 | artçı sarsıntılar sürüyor | istanbul düzce kasım depremin artçı sarsıntıları sürüyor boğaziçi üniversitesi kandilli rasathanesi deprem araştırma enstitüsü düzce büyüklüğünde artçı sarsıntı

Figure 1.2: The Preprocessed News Report

1.1.2 Wild Card Matching

Lovins [22] defines the *stemming* as a " procedure to reduce all words with the same stem to a common form, usually by stripping each word of its derivational and inflectional suffixes. "

Stemming is generally achived by means of suffix dictionaries that contain lists of possible word endings, and this approach has been applied succesfully to many languages similar to English. It is, however, less applicable to an agglutinative language such as Turkish, which requires a more detailed level of morphological techniques that remove suffixes from words according to their internal structure. Therefore, wild card procedure is used in the thesis. Wild card matching allows a term to be expanded to a group of related words. e.g., the wild card, " BAKAN* ", comprises the words of which the sequences of characters until asterisk matches with such as BAKANLIK, BAKANLAR. A special wild card list (Table 1.1) as a dictionary is created and also the most of the wild card words, derived from inflexional suffixes, resemble the stemming. Stemming procedure is to reduce all words with the same root to a common form, usually by stripping each word of its derivational and inflexional suffixes. In wild card procedure, it is not a requirement to reduce with the same root, generally, a character or a derivational suffix can remaine beside the root. However, the wild card words, derived from derivational suffixes, represent the basic difference between the wild card procedure and the stemming.

ALDA*	BİTT*	DÖNM*	GÖRÜ*
ALMA*	BULAC*	GİD*	KURTULM*
ALMIŞ*	BULM*	GİT*	KURTULA*
ALSIN*	BULD*	GİRE*	KURTULD*
CEZA*	TERÖR*	EĞİT*	JEO*
CENAZE*	DEPREM*	FİLM*	İHALE*
DAVA*	DEVLET*	FUTBOL*	KURUM*
DENİZ*	DUYURU*	FRANS*	TRAFİ*

Table 1.1: The Sample Words In Wild Card List

AMA	DOKUZ	EPEY*	PEK
FAKAT	BİN	BİRÇO*	TEKR*
GİBİ	MART	BÖYLE*	MİLYON*
LAZIM	SALI	DÖRT*	MİLYAR
AŞIN*	DURM*	GÖREN*	TUTM*
AŞILAC*	DÖND*	GÖRM*	TUTT*
BİTİR*	DÖNE*	GÖNDER*	TUTUL*
BİTM*	DÖNÜŞ*	GÖSTER*	UNUT*

Table 1.2: Some Sample Stopwords

1.1.3 Stopword and Keyword List

In order to provide efficiency, the evaluation of a wild card procedure, and formation of a stopword list containing non-formative words, and a keyword list is required. If a word is either a stopword (Table 1.2) or a keyword (Table 1.3), depends on some rules, the meaning of the word and the frequency of the word in the whole document. The frequency of occurence of the words in the dataset are found by a method called *term frequency* [43]. The most frequently occuring words are mainly function words such as conjunctions, postpositions, pronouns, etc, and these words are selected for inclusion in the stopword list. Furthermore, some of the large number of low-frequency Turkish words are morphological variants of very commonly occuring function words; these former words are also included in the stopword list.

The aim of the thesis is to compile a Turkish dataset in order to study in Text Categorization and to evaluate and compare the performance of the *FPTC*, *FBTC* and *k-NN* classifier algorithms on this dataset.

CEZA*	TERÖR*	EĞİT*	JEO*
CENAZE*	DEPREM*	FİLM*	İHALE*
DAVA*	DEVLET*	FUTBOL*	KURUM*
DENİZ*	DUYURU*	FRANS*	TRAFİ*
AVLANM*	KAÇT*	PATLA*	YÜKSELT*
BİRLEŞM*	KAÇIR*	SALDIR*	YÜKSELME*
BİRLEŞT*	KALKIN*	TUTUK*	YIKT*
ÇEKİLD*	OYNA*	VURUL*	YIKIL*

Table 1.3: Some Sample Keywords

1.2 Classifiers

Many classification algorithms, most of which are in fact machine learning algorithms, have been used for text categorization. A growing number of statistical learning methods have been applied to text categorization problem in recent years including regression models [41], nearest neighbor clasifiers [42], Bayesian probabilistic classifiers [20, 25], decision trees [20, 25], inductive rule learning algorithms [1, 4, 28] and neural networks [27].

Text Categorization is the assignment of texts to *one* or *more* of a pre-existing set of categories, on the other hand, Text Classification is the assignment of texts to *only one* of a pre-existing set of categories. In classification, given a set of classification labels C, and set of training examples E, each of which has been assigned one of the class labels from C, the system must use to predict the class labels of previously unseen examples of the same type [23].

A classifier makes a YES/NO decision for each category and if the classifier is able to produce a ranking list of m ($m > 2$) categories for each document as k-NN classifier, it is also used in Text Categorization. Given an arbitrary input document, the k-NN classifier ranks its nearest neighbor among the training documents, and uses the categories of the k top-ranking neighbors to predict the categories of the input document. The similarity score of each neighbor document to the new document being classified is used as the weight of each of its categories, and the sum of category weights over the k nearest neighbors are used for category ranking [38].

This section contains a brief overview about classifiers, namely k-NN and FPTC, that are applied on A.A. Dataset in order to evaluate the performance of the dataset.

1.2.1 k-NN Classifier

Experiments regarding those works give promosing results. However, most of algorithms are not scalable with the size of vocabulary (feature set), which is expressed in the order of tens of thousands. Here, each feature is a keyword and this requires reduction of feature set or training set in such a way that the accuracy would not degrade [12].

Among those algorithms, k-NN that is the nearest neighbor classifier and the most accurate and simplest one. It is based on the assumption that the most similar an unclassified instance should belong to the same class as the most similar instance in the training instance. To measure the similarity between two instances, several distance metrics have been proposed by Salzberg [31], of which the Eucladian distance metric is the most common.

k-NN is also scalable with the size of the feature set. In other words, it can be used to classify the documents having large feature sets while most of the algorithms can not be used because their space problem with those datasets. The k-NN algorithm is based on the idea that the less the distance of the two instances in the space, more similarity between them. Therefore, it finds the k nearest instances in the instance space and assigns the category which is among these k instances as the category of a tested instance. However, since it requires calculating the distance of the tested instance to all other instances in the training set, it is very inefficient in terms of time. Another major drawback of the similarity measure used in k-NN is that it uses all features in computing distances. In many document datasets, only smaller number of the total vocabulary may be useful in categorizing documents. A possible approach to overcome this problem is to learn weights for different features (or words in document data sets) [14].

1.2.2 Feature Projection Text Classifier

FPTC is another nearest neighbor algorithm that is developed to make kNN more time efficient [12]. It is an extension of the kNN algorithm and based on the idea of representing training instances as their projections on each feature dimension. During its training, it makes a prediction for each feature in all of the training documents. These predictions also give information about fruitfulness of a feature for classifying test instances. During its testing, the majority vote of each individual feature specifies the category of a test instances. Since the time complexity of *FPTC* algorithm is proportional to feature size of each training instance and is independent of the size of training set, it is more efficient than k-NN in terms of time.

1.3 Outline of the Thesis

In the next chapter, we present an overview of previous works regarding datasets and classifiers. In Chapter 3, the algoritms, which are used in the thesis for evaluation of dataset, are discussed and in Chapter 4, preprocessing work for Anadolu Agency Dataset is presented. The detailed description of characteristic properties of the methods are given in these chapters. Empirical evaluations of k-NN and FPTC algorithms, and the performance of them on the dataset are shown in Chapter 5, and the final chapter presents a summary of the results obtained from the experiments in the thesis. Also an overview of possible extensions to the work presented here is given as future work.

Chapter 2

Overview of Datasets and Classifiers

Much progress has been made in the past 10-15 years in the area of text categorization and in applying machine learning to text categorization. Text Categorization is at the meeting point between *ML* and *IR*, since it applies ML techniques for IR purposes. Many existing text categorization systems share certain characteristics. Namely, they all use induction as the core of learning classifiers. Moreover, they require a text representation step that turns textual data into learning examples. This step involves both IR and ML techniques. It is often difficult to detect statistically significant differences in overall performance among several of better systems whether one is employing knowledge engineering or supervised machine learning. One often finds comparisons being made on the basis of fractions of percentage point difference in some performance metric. Many methods, quite different in the technologies used, seem to perform about equally well overall [19].

To study in text categorization, one needs a pool of training data from which samples can be drawn, and a classification system against which the effects of different systems can be tested and compared [39]. On the other hand, the most serious problem in text categorization is the lack of standard data collections. Even if a common collection is chosen, there are still many ways to introduce inconsistent variations.

In this chapter, some datasets and classifiers which are the most frequently used in *Text Categorization*, are reviewed. In the first section, we review classifiers including binary and m-ary classifiers. In the second section, the most commonly used dataset collections in Text Categorization are discussed.

2.1 Classifiers

Many classification algorithms, most of which are in fact machine learning algorithms, have been used for text categorization. A growing number of statistical learning methods have been applied to text categorization problem in recent years including regression models [41], nearest neighbor clasifiers [42], Bayesian probabilistic classifiers [20, 25], decision trees [20, 25], inductive rule learning algorithms [1, 4, 28] and neural networks [27].

This section briefly overview about classifiers dividing into two main types: Indepensent binary classifiers and m-ary classifiers.

2.1.1 Binary Classifiers

Independent binary classifier makes a *YES/NO* decision for each category, independently from its decisions on other categories. The best-known binary classifiers, Construe, Decision Tree, Naive Bayes, Neural Networks, DNF, Rocchio and Sleeping Experts, are briefly discussed in the following section.

2.1.1.1 CONSTRUE

Construe is an expert system developed at Carnegie Group and the earliest system evaluated in Reuters Corpus [15]. In spite of setting a landmark in Text Categorization research, Construe design is known to be an expensive and time consuming task, since it is one of the hand-crafted knowledge engineering systems [28].

Promising results were reported on a small subset of Reuters corpus by

Yang [41]. A major difference between the *CONSTRUE* approach and the other methods is the use of manually developed domain-specific or application-specific rules in the expert system. Adopting *CONSTRUE* to other application domains would be costly and labor-intensive.

2.1.1.2 Decision Tree

Decision Tree is a well-known machine learning approach to automatic induction of classification trees based on training data [23]. A decision tree is constructed for each category using the recursive partitioning algorithm with information gain splitting rule. A probability is maintained at each leaf rather than a binary decision. Applied to text categorization, decision tree algorithms are used to select informative words based on an information gain criterion, and predict categories of each document according to the occurence of word combinations in the document. Evaluation results of decision tree algorithms on the Reuters Text Categorization collection were reported in [20].

C4.5 classifier is one of the most known text categorization Decision Tree algorithm which uses divide-and-conquer approach. This method was first developed as an extension of ID3 (Information Dichotomizer 3) by Quinlan. It progressed over several years and is now known as C4.5.

2.1.1.3 Neural Networks

Modern **Neural Networks** are descendants of the perceptron model and the least mean square (LMS) learning systems of the 50s' and 60s'. The perceptron model and its training procedure was presented for the first time by Rosemblatt and the current version of LMS by Widrow and Hoff. The simplest perceptron is a network that has an output node and an input layer that contains two or more nodes. The node in the output layer is connected to all the nodes of the input layer. The perceptron is a device that decides whether an input pattern belongs to one of two classes. The mathematical model of the perceptron corresponds to a linear discriminant.

There are two kinds of learning algorithms that can be used for training a neural network: supervised and unsupervised learning. In supervised learning, a set of examples that includes the set of input features and the expected output for each example is used. It is called supervised because during the training phase the weights of the network are adjusted until its output is closed to desired output. Backpropagation is the most prominent method of this approach. In unsupervised learning, only the value of the input features is in the hand, and the network performs a clustering or association procedure to learn the classes that are present in the training set. Examples of unsupervised neural networks are Kohonen networks and Hopfield networks [29].

As a review about Neural Network, the earliest works tried to apply feed-forward algorithms and represent the three basic elements of information retrieval system (documents, queries, and index terms) as individual layers in the neural network. The other important category of neural network applications involves more specific tasks such as conceptual clustering, document clustering and concept mapping. More extensive resarch about Reuters categorization were reported by Wiener [27].

2.1.1.4 Naive Bayes Classifier

Naive Bayes probabilistic classifiers are also commonly used in Text Categorization. The basic idea is to use the joint probabilities of words and categories to estimate the probabilities of categories in a given document. That is, Bayes Theorem is used to estimate the probability of category membership for each category and each document. Probability estimates are based on the co-occurence of categories and the selected features in the training corpus, and some independence assumption.

The Bayesian classifier estimates the *log* probability that the essay belongs to the class of "good" documents, $log(P(C|Doc))$, as follows:

$$log(P(C)) + \sum_{\mathbf{i}} \begin{cases} log \quad (P(A_i|C)/P(A_i)) \\ \qquad if\ the\ test\ doc\ has\ feature\ A_i \\ log \quad (P(\bar{A}_i|C)/P(\bar{A}_i)) \\ \qquad if\ the\ test\ doc\ does\ not\ have\ A_i \end{cases}$$

Where *P(C)* is the prior probability that any document is in Class C, the class of "good" documents, $P(A_i|C)$ is the conditional probability of a document having feature A_i given that the document is in class C, $P(A_i)$ is the prior probability of any document containing feature A_i, $P(\bar{A}_i|C)$ is the conditional probability that a document does not have feature A_i given that the document is in class C, and $P(\bar{A}_i)$ is the prior probability that a document does not contain feature A_i.

The *Naive* part of such a model is the assumption of word independence. The simplicity of this assumption makes the computation of the Naive Bayes classifier far more efficient than the exponential complexity of non-naive Bayes approaches because it does not use word combinations as predictors. Evaluation results of Naive Bayes classifier on Reuters were reported by Lewis Ringuette [20] and Moulinier [25], respectively. And also there exists an extensive research ,reported by Larkey [18].

Rainbow is a Naive Bayes classifier for text classification tasks [23], developed by Andrew McCallum at CMU. It estimates the probability that a document is a member of a certain class using the probabilities of words occuring in documents of that class independent of their context. By doing so Rainbow makes the naive independence assumption [9].

More precisely, the probability of document d belonging to class C is estimated by multiplying the prior probability $P(C)$ of class C with the product of the probabilities $P(w_i|C)$ that the word w_i occurs in documents of this class. This product is then normalized by the product of the prior probabilities $P(w_i)$ of all words.

$$P(C|d) = P(C) \prod_{i=1}^{n} \frac{P(w_i|C)}{P(w_i)} \tag{2.1}$$

PropBayes algorithm [20] uses Bayes' rule to estimate the category assignment probabilities, and then assigns to a document these categories with high probabilities. PropBayes estimates $P(C_j = 1|D)$, the probability that a category C_j should be assigned to a document, based on the prior probability of a category occuring, and the conditional probabilities of particular words occuring in a document given that it belongs to a category. For tractability, the assumption is made that probabilities of word occurences are independent of each other, though this is often not the case. Detailed research and comparison of PropBayes with Decision Tree algorithms are reported by Lewis [20].

2.1.1.5 Inductive Rule Learning in Disjunctive Normal Form

Disjunctive Normal Form (DNF) algorithms express their results as a logical formula in disjunctive normal form. DNF was tested in the *RIPPER* and *CHARADE* systems [4, 28], respectively. DNF rules are of equal power of decision trees in machine learning theory. Emprical results for the comparison between DNF and decision tree approaches, however, are rarely available in text categorization researches, except in an indirect comparison by Apte [1].

RIPPER is an algorithm for inducing classification rules from a set preclassified examples. The user provides a set of examples, each of which has been labeled with the appropriate class. Ripper then looks at the examples and finds a set of rules that will predict the class of unseen examples.

More precisely, RIPPER builds a ruleset by repeatedly adding rules to an empty ruleset until all positive examples are covered. Rules are formed by first splitting the training data into two sets, a "growing set" and a "prunning set". And then greedily adding conditions to the ancedent of a rule with an empty ancedent until no negative examples are covered; after such a rule is found, the rule is simplified, by greedily deleting conditions so as to improve the rule's performance on the "prunning' examples. In this phase of learning, different *ad hoc* heuristic measures are used to guide the greedy search for new conditions, and the greedy search for simplification.

After a ruleset is thus constructed, an "optimization" phase modifies the

ruleset so as to reduce its size and improve its fit to the training data. Each pass of the optimization involves looping over each rule R in the constructed ruleset, and attempting to construct a replacement for R that improves performance of the entire ruleset. To construct candidate replacements, a strategy similar to the one used to construct rules in the covering phase is used: a rule is grown, and then simplified, with the goal of simplification being now to reduce the error of the total ruleset on another held-out "prunning" set. There exists an extensive research in the literature, reported by Cohen [4, 5].

k-**DNF** learners are symbolic ML algorithms, that express the learned concepts as formula in disjunctive normal form; each disjunct has at most k literals. Production rule learners, such as **CHARADE**, are typical k-**DNF** learners. **CHARADE** is said to construct consistent descriptions of concepts, ie., a description is generated when all examples covered by this description belong to the same concept. CHARADE relies on the simultaneous exploration of the description space and the instance space. The description space D is defined as the power-set of the set of descriptors, while the instance space is the power-set of the learning set. The inductive process combines descriptions in D, beginning with simple descriptions. The algorithm stops when the instance space has been exhausted. This strategy enables redundant learning, since an example can be covered several times. Such learners are not noise-resistant. However, most ML techniques provide some means to take noise into account. An extensive research about comparison of Charade with other classification methods are available in the literature [24].

2.1.1.6 Rocchio

The **Rocchio** algorithm is a *batch* algorithm. It produces a new weight vector $\mathbf{w}$ from an existing weight vector $\mathbf{w_1}$ and a set of training examples [21]. However, Rocchio is a classic vector-space model method for document routing or filtering in information retrieval. Applying it to text categorization, the basic idea is to construct a prototype vector per category using a training set of documents. Given a category, the vectors of documents belonging to this category are given a positive weight, and the vectors of remaining documents are given a

negative weight. By summing up those positively and negatively weighted vectors, such a prototype vector is called *centroid* of the category. This method is easy to implement and efficient in computation, and has been used as a baseline in several evaluations [4, 21]. A potential weakness of this method is the assumption of one centroid per category, and consequently, Rocchio does not perform well when the documents belonging to a category naturally form separate clusters [38].

2.1.1.7 Sleeping Experts (EXPERTS)

EXPERTS are on-line learning algorithms recently applied to text categorization. It is based on a new framework for combining the "advice" of different "experts" (or in another word the predictions of several classifiers) which has been developed within the computational learning community over the last several years. Prediction algorithms in this framework are given a pool of fixed "experts" -each of which is usually a simple, fixed classifier- and build a master algorithm, which combines the classifications of the experts in some manner. Building a good master algorithm is thus a matter of finding an appropriate weight for each of the experts. The examples are fed one-by-one to the master algorithm, which updates the weight of different experts based on their prediction on that example.

On-line learning aims to reduce the computation complexity of the training phase for large applications. EXPERTS updates the weights of n-gram phrases incrementally.

2.1.2 m-ary Classifiers

M-ary classifier typically uses a shared classifier for all categories, producing a ranked list of candidate categories for each test document, with a confidence score for each candidate. The best-known m-ary classifiers, Linear Least Squares Fit (LLSF), Word, k-NN and k-NNFP, are briefly discussed in the following section.

2.1.2.1 Linear Least Squares Fit

LLSF is a mapping approach developed by Yang [38].A multivariate regression model is automatically learned from a training set of documents and their categories. The training data are represented in the form of input/output vector pairs where the input vector is a document in the conventional vector space model (consisting of weights for words), and output vector consists of categories (with binary weights) of the corresponding document. By solving a *LLSF* on the training pairs of vectors, one can obtain a matrix of word-category regression coefficients. The matrix defines a mapping from an arbitrary document to a vector of weighted categories. By sorting these category weights, a ranked list of categories is obtained for the input document.

2.1.2.2 Word

Word is a simple, non-learning algorithm which ranks categories for a document based on word matching between the document and category names. The purpose of testing such a simple method is to quantitatively measure how much of improvement is obtained by using statiscal learning compared to a non-learning approach. The conventional vector space model is used for representing documents and category names (each name is treated as a bag of words) and the SMART [30] system is used as the search engine.

2.1.2.3 k-Nearest Neighbor

Given an arbitrary input document, the system ranks its nearest neighbors among the training documents, and uses the categories of k top-ranking neighbors to predict the categories of the input document. There are two main methods for making a prediction training documents: majority voting and similarity score summing. In major voting, a category gets only one vote for each instance of that category in the set of k top-ranking nearest neighbors. However, the most similar category is the one that gets the highest score of

votes. In the latter, each category gets a score equal to the sum of the similarity scores of the instances of that category in the k top-ranking neighbors. The most similar category is the one with the highest similarity score sum. In other words the less the distance of the two instances in the space, more similarity between them. The similarity score of each neighbor document to the new document being classified is used as the weight of each of its categories, and the sum of category weights over the k nearest neighbors are used for category ranking. The similarity value between two instances is the distance between them based on a distance metric. In general, the *Eucladian Distance Metric* is the most commonly used.

2.1.2.4 k Nearest Neighbor Feature Projection

k-NNFP technique is a variant of k-NN method [30]. The most important characteristic of **k-NNFP** technique is that the training instances are stored as their projections on each feature dimension and distance between two instances is calculated according a single feature. This allows the classification of a new instance to be made much faster than k-NN. Since each feature is evaluated independently if the distribution of categories over the data set is even, votes returned for the irrelevant features will not adversely affect the final prediction. That is, the voting mechanism reduces the negative effect of possible irrelevant features in classification. The more detailed expression is presented in the next chapter.

2.2 Data Collections

Dataset selection is important for both the effectiveness and the efficiency of statistical text categorization. That is, we want a training set which contains sufficient information for example-based learning of categorization, but is not too large for efficient computation. The latter is particularly important for solving large categorization problems in practical databases.

The amount of available training data is often nearly infinite [39]. But, the

Version	(prepared by)	UniqCate	Train	Test	(Labelled TestDocs)
Version 1	(CGI)	182	21450	723	(80%)
Version 2	(Lewis)	113	14704	6746	(42%)
Version 2.2	(Yang)	113	7789	3309	(100%)
version 3	(Apte)	93	7789	3309	(100%)
Version 4	(PARC)	93	9610	3662	(100%)

Table 2.1: Different versions of Reuters

most serious problem in Text Categorization evaluation is the lack of standard data collections. Even if a common collection is chosen, there are still many ways to introduce inconsistent variations [38].

Yang focus on the following questions regarding effective and efficient learning of text categorization [38]:

- Which training instances are most useful? Or, what sampling strategies would globally optimize text categorization performance?
- How many examples are needed to learn a particular category?
- Given a real-world problem, how large a training sample is large enough?

In the following sections, some of the most commonly used data collections in Text Categorization are reviewed.

2.2.1 Reuters

Reuters is the most commonly used collection for text categorization evaluation in the literature. The Reuters corpus consists over 20000 Reuters newswire stories in the period between 1987 to 1991. The original corpus (Reuters-22173) was provided by the Carnegie Group Inc. and used to evaluate their *CONSTRUE* system in 1990 [15]. Several versions have been derived from this corpus by varying the documents in the corpus, the division between the training and test set, and the categories used for evaluation. Table 2.1 summarizes these versions [38].

Reuters version 2 (also called Reuters-21450), prepared by Lewis [20], contains all of the documents in the original corpus (Version 1) except the 723 test documents. The documents are split into two chronologically contiguous chunks; the early one is used for training, and the later one for testing. A subset of 113 categories were chosen for evaluation. One peculiarity of Reuters-22450 is the inclusion of a large portion of unlabeled documents in both the training (47%) and test (58%) test sets. It is observed by Yang [38] that on randomly tested documents, in many cases, the documents do belong to one of those 113 categories but happen to be unlabelled. And Carnegie Group confirmed that Reuters does not always categorize all of their news stories. However, it is not known exactly how many of the unlabeled documents should be labelled with a category.

Yang created a new corpus from Reuters 2, called *Reuters version 2.2*, in order to facilitate an evaluation of the impact of these unlabeled documents on text categorization. The only difference among them is that all of the unlabeled documents have been removed.

Reuters version 3 was constructed by Apte for their evaluation of the SWAP-1 by removing all of the unlabeled documents from the training and test sets and restricting the categories to have training set frequency of at least two [1, 38] Fig 2.1.

Reuters version 4 was constructed by the research group at Xerox PARC, and was used for the evaluation of their neural network approaches [27]. This version was drawn by from Reuters version 1 by eliminating the unlabeled documents and some rare categories. Instead of taking continuous chunks of documents for training and testing, it slices the collection into many small chunks that do not overlap temporally. Those subsets are numbered, and the odd-numbered chunks are used for training and the even subsets are used for testing [38].

.I 626
.C
acq 1
.T
KUWAIT INCREASES STAKE IN SIME DARBY.
.W
KUALA LUMPUR, April 11 - The Kuwait Investment Office (KIO) has increased its stake in ¡Sime Darby Bhd¿ to 63.72 mln shares, representing 6.88 pct of Sime Darby's paid-up capital, from 60.7 mln shares, Malayan Banking Bhd ¡MBKM.SI¿ said. Since last November, KIO has been aggressively in the open market buying shares in Sime Darby, a major corporation with interests in insurance, property development, plantations and manufacturing. The shares will be registered in the name of Malayan Banking subsidiary Mayban (Nominees) Sdn Bhd, with KIO as the beneficial owner.

.I 631
.C
interest 1
.T
YIELD RISES ON 30-DAY SAMA DEPOSITS.
.W
BAHRAIN, April 11 - The yield on 30-day Bankers Security Deposit Accounts issued this week by the Saudi Arabian Monetary Agency (SAMA) rose by more than 1/8 point to 5.95913 pct from 5.79348 a week ago, bankers said. SAMA decreased the offer price on the 900 mln riyal issue to 99.50586 from 99.51953 last Saturday. Like-dated interbank deposits were quoted today at 6-3/8, 1/8 pct – 1/8 point higher than last Saturday. SAMA offers a total of 1.9 billion riyals in 30, 91 and 180-day paper to banks in the kingdom each week.

Figure 2.1: The Reuters Version 3 Dataset

2.2.2 Associated Press

The document of 371,454 items which appeared on the **Associated Press (AP)** newswire between 1988 and early 1993 were divided randomly into a training set of 319,463 documents and a test set of 51,991 documents. The headlines are an average of 9 words long, with a total vocabulary is 67,331 words. No preprocessing of the text was done, except for converting all words to lower case and remove punctuation. Word boundaries were defined by whitespace. Titles were used, rather than the full text of the items, to minimize computation.

Categories to be assigned were based on the "keyword" from the "keyword slug line" present in each *AP* item. The keyword is a string of up to 21 characters indicating the content of the item. While keywords are only required to be identical for updated items on the same news story, in practice there is a considerable reuse of keywords and parts of keywords from story to story and year to year, so they have some aspects of a controlled vocabulary [10].

2.2.3 OHSUMED (Medline)

OHSUMED is a bibliographical document collection developed by William Hersh and collegues at the Oregon Health Sciences University. It is a subset of the *Medline* database consisting of 384,566 documents were manually indexed using subject categories (Medical Subject Headings or MESH) in the National Library of medicine. There are about 18,000 categories defined in the MESH and 14,321 categories present in the OHSUMED document collection. The average length of a document is 167 words. On average 12 categories are assigned to each document Fig 2.2.

In some sense, the OHSUMED corpus is more difficult than Reuters, because the data are more "noisy". That is, the word / category correspondences are more "fuzzy" in OHSUMED. Consequently, the categorization is more difficult to learn for a classifier [43].

2.2.4 USENET

Most work in classification has involved articles taken off a newswire or from a medical database. In these cases, correct topic labels are chosen by human experts. The domain of **USENET** newsgroup postings is another interesting testbed for classification Fig 2.3. The "labels" are just the newsgroups to which the documents were originally posted. Since users of the Internet must make this classification decision everytime they post an article, this is a nice "real life" application of text categorization [35].

Contributors to *USENET* often vary in their use of terminology, stray from

the topic, or use unusual language. All of these qualities tend to make subject-based classification tasks from USENET more difficult than those of a comparable size from Reuters [33].

2.2.5 DIGITRAD

DIGITRAD is a public domain collection of 6,500 folk song lyrics. To aid searching, the owners of DigiTrad have assigned to each song one or more keywords from a fixed list. Some of these keywords capture information on the origin or style of the songs (e.g. "Irsh" or "British) while others related to subject matter (e.g. "murder" or "marriage"). The latter type of keywords served as the basis for the classification tasks in the studies. The texts in DigiTrad make heavy use of metaphoric, rhyming unusual and archaic language. Since the lyrics do not often explicity state what a song is about, it makes the categorization difficult for a categorizer.

.I 274274
.C
Adult 1; Case-Report 1; Cysts 1; Ear-Diseases 1; Ear,-External 1; Human 1; Male 1
.T
Pseudocyst of the auricle. Case report and world literature review
.W
We treated a patient with pseudocyst of the auricle and reviewed the 113 cases previously published in the world literature. Pseudocyst of the auricle is an asymptomatic, noninflammatory cystic swelling that involves the anthelix of the ear, results from an accumulation of fluid within an unlined intracartilaginous cavity, and occurs predominantly in men (93% of patients). Characteristically, only one ear is involved (87% of patients), and the lesion is usually located within the scaphoid or triangular fossa of the anthelix. Previous trauma to the involved ear is uncommon. The diagnosis may be suggested by the clinical features, and analysis of the aspirated cystic fluid and/or histologic examination of a lesional biopsy specimen will confirm the diagnosis. Therapeutic intervention that maintains the architecture of the patient's external ear should be used in the treatment of this benign condition.

.I 274230
.C
Accidents 1; Adolescence 1; Adult 1; Aged 1; California 1; Case-Report 1; Cause-of-Death 1; Child 1; Child,-Preschool 1; Coronary-Disease 1; Emergency-Service,-Hospital 1; Female 1; Heart-Diseases 1; Homicide 1; Human 1; Infant 1; Male 1; Middle-Age 1; Retrospective-Studies 1; Suicide 1; Survival-Rate 1
.T
Cause of death in an emergency department
.W
A retrospective review was done of 601 consecutive emergency department deaths. Nontrauma causes accounted for 77% of the deaths and this group had an average age of 64 years and a male to female ratio of 1.9:1. Trauma caused 23% of the fatalities and this group had a younger average age of 29 years and a male to female ratio of 4.6:1. The most common causes of nontrauma death were sudden death of uncertain cause (34%), coronary artery disease (34%), cancer (5%), other heart disease (4%), chronic obstructive lung disease (3%), drug overdose (3%), and sudden infant death syndrome (2%). The most common causes of trauma death were motor vehicle accidents (61%) and gunshot wounds (16%). The overall autopsy rate was 40%. Death certificates were often in error

Figure 2.2: The Original OHSUMED Dataset

Subject: a-life graduate studies?
Date: Sun, 19 Mar 2000 13:23:46 -0500
From: "fish" fish@7cs.net
Newsgroups: comp.ai.alife

Hi all, I'm looking for a multidisciplinary graduate program in a-life and was wondering if the newsgroup had any recommendations. I am currently teaching 3D character animation, intro to programming, and courses in game development and VRML at the Savannah College of Art Design www.ca.scad.edu Thanks in advance.
greg johnson gjohnson@scad.edu

Subject: The Sims... anyone?
Date: Fri, 24 Mar 2000 03:42:01 -0600
From: jorn@mcs.com (Jorn Barger)
Organization: The Responsible Party (conservative left)
Newsgroups: comp.ai.games,comp.ai.alife
Did I already miss the big, excited thread about the Sims? I read where it's the seller, so why aren't people talking about it on cag and caa? Has anyone reverse-engineered a list of the 'semantic' variables? [Semi-unrelated issue that was what I really wanted to ask about when I peeked in:] Do any social sims use a model where, before any act, they consider each other actor, and how the proposed act will affect them? I'm thinking it's like 'how much will this entangle our karmas ?' To the Sirens first shalt thou come, who bewitch all men... I edit the Net: URL:http:www.robotwisdom.com "...frequented by the digerati"
The New York Times

Figure 2.3: The Original USENET Messages

Chapter 3

Text Categorization Algorithms Used

Many machine learning algorithms have been applied to text categorization as briefly described in *Chapter 2*. And most of them give promosing results, but some of them are not scalable with the size of feature set, which is expressed in order of tens of thousands. Scalability is a fundemantal problem in text categorization. Since it requires reduction of feature set or training set in such a way that the accuracy would not degrade. However, the *m-ary* algorithms like k-NN can be used with large set of the features compared to the other existing methods.

As we mentioned in *Chapter 1*, the motivation behind the work of the thesis is to evaluate the Turkish language. Turkish is an agglutinative language, therefore it requires text processing techniques different than English and similar languages on text categorization. We apply two algorithms on the dataset, namely FPTC and k-NN classifiers for evaluation and comparison.

In this chapter we examine the description and complexity of algorithms, applied on the dataset. The description and complexity of FPTC algorithm is described in the first section. And in the second section, k-NN algorithm is discussed.

3.1 The FPTC Algorithm

FPTC algorithm [12] is a variant of k-NN and a non-incremental algorithm that is all training instances are taken and processed at once. The main characteristic of the algorithm is that instances are stored as their projections on each feature dimension. If the value of a training instance is missing for a feature, that instance is not stored on that feature. However, another characteristic of the algorithm is that distance between two instances is calculated according to a single feature.

The distance between the values on a feature dimension is computed using $diff(f, x, y)$ metric as foolows:

$$\mathbf{diff}(\mathbf{f}, \mathbf{x}, \mathbf{y}) = \begin{cases} |x_f - y_f| & \text{if f is linear} \\ 0 & \text{if f is nominal and } \mathrm{x}_f = y_f \\ 1 & \text{if f is nominal and } \mathrm{x}_f \neq y_f \end{cases}$$

However, since each feature is processed separately, this metric does not require normalization of feature values. If there are f features, this method returns $f \times k$ votes whereas k-NN method returns k votes.

A preclassification, separately on each feature, is performed in order to classify an instance. For a given test instance t and feature f, the preclasification for $k = 1$ will be the class of the training instance whose value on feature f is the closest to that of the t. For a larger value of k, the preclassification is a bag (multiset) of classes of the nearest k training instances. In other words, each feature has exactly k votes, and gives these votes for the classes of the nearest training instances. For the final classification of the test instance t, the preclassification bags of each feature are collected using bag union. Finally, the class that occurs most frequently in the collection bag is predicted to be the class of the test instances. In other words, each feature has exactly k votes, and gives these votes for the classes of the nearest training instances [11].

All the projections of training instances on linear features are sorted in memory as sorted values. In Figure 3.1, the function *kBag(f,t,k)*, which returns a bag of size k containing the classes of the k nearest training instances to the

```
classify(t,k)
/* t:test instance, k:number of neighbors */
[1]    begin
[2]       for each class c
[3]          vote[c] = 0

[4]       for each feature f
[5]          /* put k nearest neighbors of test instance t on feature f into Bag */
[6]          Bag=kBag(f,t,k)
[7]          for each class c
[8]             vote[c] = vote[c] + count[c,Bag];

[9]       prediction= UNDETERMINED /* class 0 */
[10]      for each class c
[11]         if vote[c] > vote[prediction] then
[12]            prediction=c

[13]      return(prediction)
[14]   end.
```

Figure 3.1: Classification in the FPTC Algorithm

instance t on feature f, computes the votes of a feature. As mentioned in Equation 3.1, distance between the values on a feature dimension is computed by using $diff(f, x, y)$ metric. Note that the bag returned by *kBag(f,t,k)* does not contain any UNDETERMINED class as long as there are at least k training instances whose f values are known. Then, the number of votes for each class is incremented by the number of votes that a feature gives to that class, which is determined by the *count* function. The value of *count(c,Bag)* is the number of occurences of class c in bag *Bag*.

There are two methods for finding the most similar instance: majority voting and similarity score summing.

In major voting, a category gets one vote for each instance of that category in the set of k top-ranking nearest neighbors. Then the most similar category is the one that gets the highest amount of votes. In *similarity score summing*, each category gets a score equal to the sum of the similarity scores of the

instances of that category in the k top-ranking neighbors. The most similar category is the one with the highest similarity score sum.

For an irrelevant feature f, the number of occurences of a class c in a bag returned by *kBag(f,t,k)* is proportional to the number of instances of class c in the training set. If majority voting is used in FPTC algorithm and the categories are equally distributed over the test instances and training set, then the votes of an irrelevant feature will be equal for each class, and the final prediction will be determined by the votes of the relevant features. If the distribution of the categories over the data set is not equally, then the votes of an irrelevant feature will be the highest vote for the most frequently occuring class.

If similarity score summing is used and the categories are equally distributed over the test instances then the similarity score sum of an irrelevant feature will be equal for each category and it will not be effective in the prediction phase. However, if the categories are not evenly distributed then the similarity score sum of an irrelevant feature will be higher for most frequently occuring class.

The FPTC algorithm handles unknown feature values by not taking them into account. If the value of a test instance for a feature f is missing, then feature f does not participate in the voting for that instance or in short, missing values are simply ignored. Needless to say that this is a natural approach regarding the real life, since if nothing is known about a feature, ignoring that feature is a normal behavior . Final voting is done between the features for which the test instance has a known value. That is, unknown feature values are simply ignored.

As mentioned before, because of storing all the training instances in the memory, the space required for training with m instances on a domain with n features is directly proportional to $m \times n$.

All instances are not only stored on each feature dimension as their feature projections in the training phase, but also sorted once at the end. Let a dateset containing m instances and n features, the training time *complexity* of

the FPTC algorithm is $\mathbf{O}(n \times m \times log\ m)$.

The $kBag(f, t, k)$ function, to determine the votes of a feature, first finds the nearest neighbor of t on f and then next $k-1$ neighbors around the nearest neighbor. The time complexity of this process is $\mathbf{O}(log\ m + k)$. Since $m >> k$, the time complexity of $kBag$ is $\mathbf{O}(log\ m)$. The final classification requires the votes of each of n features. Therefore, the classification time complexity of the FPTC algorithm is $\mathbf{O}(n \times log\ m)$ [11].

3.2 k-NN Algorithm

The **k-NN** classifier [7] classifier is the basis of many *lazy learning* algorithm and it is sure that k-NN is purely lazy. Purely lazy learning algorithms generally are characterized by three behaviors: [2]

1. *Defer:* They store all training data and defer processing until queries are given that require reply.
2. *Reply:* Qeries are answered by combining the training data, typically by using a *local learning approach* in which (1) instances are defined as points in a space, (2) a similarity function is defined on all pairs of these instances, (3) a prediction function defines an answer to be a monotic function of query similarity.
3. *Flush:* After replying to a query, the answer and intermediate results are discarded.

As a result, we can say that k-NN simply stores the entire training set and postpones all effort towards inductive generalization until classification time. k-NN generalizes by retrieving the k least distance (most similar) instances of a given query and predicting their weighted-majority class as the query's class. Therefore, it is doubtless that the quality of k-NN prediction depends on which instances are assumed least distant, and which is determined by its distance function.

In the basic method, learning appears almost trivial-one simply stores each training instance, which is represented as a set of feature-value pair, in memory. The power of the process comes from the retrieval process. Given a new test instance, one finds the stored training case that is nearest according to some distance measure, notes the class of the retrieved case, and predicts the new instance will have the same class.

Training:
[1] $\forall \mathbf{x}_t \in$ Training Set
[2] Store $\mathbf{x}_t$ in memory

Querying:
[1] $\forall \mathbf{x}_q \in$ Query Set
[2] $\forall \mathbf{x}_t \{\mathbf{x}_t \neq \mathbf{x}_q\}$: Calculate $Similarity(\mathbf{x}_q, \mathbf{x}_t)$
[3] Let $Similars$ be set of k most similar instances to $\mathbf{x}_q$ in Training Set
[4] Let $Sum = \sum_{\mathbf{x}_t \in Similars} Similarity(\mathbf{x}_q, \mathbf{x}_t)$
[5] Then return the categories of instances in Similar, in decreasing order by the number of times the category is seen in Similar.

Figure 3.2: The k Nearest Neighbor Regression

There is a variety of k nearest neighbor classifier approaches in the literature. Stanfill and Waltz [34] introduced the Value Added Metric (VAD)) to define similarity when using symbolic-valued features. Kelly and Davis [17] introduced the *weighted* k-NN algorithm and a recent work by Salzberg [32] has given the best case results or the nearest neighbor learning. An experimental comparison work the NN and Nested Generalized Examplers is presented by Wettschereck and Dietterich [36]. The algorithm, shown in Figure 3.2, is the simplest k nearest neighbor classifier approach. For a given query instance, k nearest (similar) training instances are determined by using the *Cosine Similarity* function.

k-NN classifies a new instance by a majority voting among its k ($k > 1$) nearest neighbors using some distance metrics. If the attributes of the data are equally important, this algorithm is quite effective. However, it can be less effective when many of the attributes are misleading or irrelevant to classification. Because of the sensitivity to the number of irrelevant features, the accuracy of k-NN can be degraded.

In spite of sensitivity to the number of irrelevant features, k-**NN** algorithm has several important properties which make suitable for our experiments:

1. k-NN is a m-ary classifier providing a global ranking of categories given a document. This allows a straight-forward global evaluation of per document categorization performance, i.e., measuring the goodness of category ranking given a document, rather than per category performance as is standard when applying binary classifiers to the problem [38].

2. k-NN classifier is context-sensitive in the sense that no independence is assumed between either input variables (terms) or output variables (categories). k-NN treats a document as a single point. A context-sensitive classifier makes better use of the information provided by features than a context-free classifier do, thus enabling better observation on feature selection [43].

3. k-NN is a non-parametric and non-linear classifier, that makes assumptions about the input data. Hence an evaluation using the k-NN classifier should reduce the possibility of classifier bias in the results [43].

k-**NN** classifier is intuitive and easy to understand, it learns quickly, and it provides good accuracy for a variety of real-world classification tasks. However, we know that k-NN has several weakness as the followings:

- Its accuracy degrades rapidly with the introduction of noisy data.
- Its accuracy degrades with the introduction of the irrelevant features.
- It has no ability to change the decision boundaries after storing the training data.
- It has large storage requirements, because it stores all training data in memory.
- It is slow during execution, because all of the training instances must be searched in order to classify each new input vector.

- Its distance functions are inappropriate or inadequate for applications with both linear or nominal attributes [37].

In the k-**NN** algorithm, the classification of a test instance requires the computation of its distance to m training instance on n dimensions. Therefore, the classification time complexity of the k-NN algorithm is simply $\mathbf{O(n \times m)}$ assuming $m >> k$.

Chapter 4

Preprocessing for Turkish News

Data preprocessing is the first operation on any set of data and consists of all the actions taken before the actual data analysis process start. However, it is usually a time consuming task and in many cases, is semi-automatic. Data preprocessing may be performed on the data for the following reasons:

- solving data problems that may prevent us from performing any type of analysis on the data,
- understanding the nature of the data and performing a more meaningful data analysis,
- extracting more meaningful knowledge from a given set of data.

Needless to say that identification of dataset has a crucial importance on preprocessing and in the thesis the data to be preprocessed is Turkish Anadolu Agency news reports. We had many time consuming difficulties not only because of ordinary data problems, preventing efficient use of the classifiers or which may result in generating unacceptable results, but also because of the morphological structure of Turkish language.

Unlike the main Indo-European languages, such as French, German and English, *Turkish* is an example of an agglutinative language, where words are formed by affixing morphemes to a root in order to extend its meaning or

to create other classes of words. In Turkish, the process of adding one suffix or affix to another, can result in relatively long words, which often contain an amount of semantic information equivalent to a whole English phrase, clause or sentence. Due to this complex morphological structure, a single Turkish word can give rise to a very large number of variants. Moreover, Turkish is a free constituent order language. The order of the constituents may change freely according to the discourse context and the syntactic role of the constituents is indicated by their case marking.

One of the important steps of preprocessing in natural language is parsing words and spell checking. Although many word parsers and spell checkers for English and some other languages have been developed, so far no such tool has been developed for Turkish.

The main aim of the preprocessing Turkish news reports in the thesis is not only solving data problems, such as noisy data, irrelevant or missing attributes in the dataset, but also changing the structure of data in order to prepare the data for a more efficient classification.

In order to apply text categorization on the A.A. dataset, we represent documents as feature vectors. Texts are represented using the conventional vector space model [30]. That is, each news report is represented as a feature vector whose dimensions are unique words in the dataset, and whose elements are feature weights. A feature is weighted using the term frequency technique.

In this chapter, preprocessing for Turkish news is presented. General steps, applied on the dataset during preprocessing is briefly discussed in the first section. The changing of the structure of dataset and filtering is examined in the second section. The third section presents preparation of wild card list and forming of stopword and keyword list. And the fourth section examines assignning weights to the features. Then assigning of categories is discussed in the last section.

4.1 General Steps

Text Categorization is quite difficult due to certain characteristics of the A.A. dataset: texts are not naturally represented as a feature vector; there is a large number of features; there is a large number of documents; there is a large variation in the amount of information in each document; the documents are written in Turkish language and so many contain ambiguities; the information needed to correctly categorize may be completely implicit or hidden; the documents are written by humans and so many contain errors; the documents contain a host of tokens besides words, such as numeric information, abbreviations, etc. High dimensionality and noisy data are the main characteristics that make text categorization in the A.A. dataset very challenging. Therefore, preprocessing is inevitable in order to provide efficient categorization. General steps, followed during preprocessing, are as follows:

- All words in the dataset are extracted and sorted into a vocabulary list. The number of different words in the vocabulary list is 18256.
- The words which have the same meaning, but are affixed with different suffixes, especially inflexional suffixes, are gathered as a single word by using wild card matching.
- The irrelevant and non-informative words are gathered in a stopword list. The number of stopwords is 876. After removal of stopwords, there exists 7856 words remained in the feature vector Table 4.1.
- Each document was represented with term frequency of these 7856 features in that document.

4.2 Data Filtering

To change the structure of the news reports and filtering from punctuation are the main requirements in order to make an efficient preprocessing. The original Anadolu Agency news reports Fig 4.1 have many unnecesssary bodies

ab	abc	abd*	abidjan
abit	abone*	abraham	abs
acapulco	ad	ada	adalet*
adana*	adanın	adapazarı	aday*
adedi	adem	adet*	adil
adjusting	adli*	adt	adıvar
adıyaman	aecl	aerospace	aesob
af	afet*	affa	affairs
affan	afgan*	afif	afkula
afp	afrik*	afyon	afyondazkırı
baas	babur	babür	back
badem	bademli	baden	bafra
bahadırlı	bahar	bahara*	bahardan
bahonar	bahreyn	bahtınız	bahçıvan
bahıtjan	bahşetmekten	bak	baka
bakalm	bakan*	baki	bakiyesi
baklan	baklava*	bakliyat	bako
baku	bakü	bakım*	bakınca
bakıp	bakır	bakırky	bakırkyspor
bal	balatası	baldo	balduk
cacharel	cadde*	cafer	cahil*
cahit	caizse	caja	cakarta
calif*	cam	cambaz	cami*
camp*	camspor	can	canavar*
canbal	cancer	candan	canikli
cankurt*	canlı*	cannes	canpolat
cantegril	canın*	cap	capone
caracas	cardenas	cargill	cari
darbe*	darboğazının	dardanelspor	daresi
dargel*	dargeçit	darp	darülaceze
darüşşafaka	darıca	data	dava*
davet*	david	davos	dayak*
dayalı	dayapmaya	dayara	dağ
dağc*	dağl*	dağ	dağnda
da	dbp	dcr	dechastelain
eker	eki	ekib*	ekici
ekimi*	ekin	ekip*	ekme*
eknomisini	ekolojik	ekonomi*	ekosistem
ekspertiz	ekspres	ekti*	ekvator

Table 4.1: The Sample Feature Vector

and in order to provide an efficient categorization some of them are removed or changed as follows:

- The main task in data filtering is to arrange each news report as a single line and separate each of them with a newline. Therefore, paragraphs and empty lines are removed **Fig 4.2**.

- All words in the news report are converted to lowercase and punctuation is removed. Word boundaries are defined by whitespace.
  -ANKARA'DA OKULLARA KAR TATİLİ...
  -TRAFİK KAZASI: 1 ÖLÜ...
  ankara'da okullara kar tatili
  trafik kazası 1 ölü

- Apostrophe 's' and the affixes, combined to the apostrophe, are removed.
  Ziyapaşa Bulvarı'nda yolun karşısına geçmek isteyen
  Şükrü Bulan'a (80)
  ziyapaşa bulvarı yolun karşısına geçmek isteyen
  şükrü bulan

- The abbreviation of *Anadolu Agency* is removed.
  ANKARA **(A.A)** ADANA **(A.A)**

- All tokens in parenthesis are removed.
  yolun karşısına geçmek isteyen Şükrü Bulan'a **(80)**

- All of the numeric values and information (bin, milyon) are removed.
  Ankara'da iki gündür etkili olan kar yağışı
  ankara gündür etkili kar yağışı
  07:51 04/01/00

- All words in parenthesis are removed.
  Istanbul Ticaret Odas **(ito)**

- The abbreviations and cryptogram, related with the subject, in the parenthesis are removed.
  (DA-CÜN-SRP)

- Some of the misspelled words, because of dialect differences in Turkish, are corrected.
  tırafik, etkonomik, tırapzonspor, kagıt, abiy.

- Each news report is divided into three bodies by the sign of (|):

  1. Categorization numbers are manually indexed using 78 subject categories.
  2. Headline text body which is an average of 12 words, is a short summary of news reports.
  3. Text body whose average length is 96 words, is the full description of the news report.

- Each category number boundary is defined by a whitespace Fig 4.2.

- Each category number is assigned sorted Fig 4.2. The detailed information about categories is in Section 4.1.3.

4.3 Wild Card

Wild card matching allows a term to be expanded to a group of related words. e.g., the wild card, ' DEPREM* ', comprises the words of which the sequences of characters until asterisk matches with. A special wild card list as a dictionary is created and the most of the wild card words, derived from inflexional suffixes, resemble the stemming. Stemming procedure is to reduce all words with the same root to a common form, usually by stripping each word of its derivational and inflexional suffixes. In wild card procedure, it is not a requirement to reduce with the same root, generally, a character or a suffix can be remained beside the root. However, the wild card words, derived from derivational suffixes, cause the basic difference between the wild card procedure and

ANKARA'DA OKULLARA KAR TATİLİ...

ANKARA (A.A) - Ankara'da kar yağışı nedeniyle okulların bugün tatil edildiği bildirildi.

Ankara Valiliği'nden yapılan açıklamada, Ankara'da iki gündür etkili olan kar yağışı sebebiyle merkez ilçelerinde bulunan ilköğretim, lise ve dengi okulların bugün tatil edildiği bildirildi.
(CÜN-SRP)
07:25 04/01/00

TRAFİK KAZASI: 1 ÖLÜ...

ADANA (A.A) - Adana'da meydana gelen trafik kazasında bir kişi öldü.

Alınan bilgiye göre, sürücünün kimliği ve plakası belirlenemeyen bir araç, Ziyapaşa Bulvarı'nda yolun karşısına geçmek isteyen Şükrü Bulan'a (80) çarparak, ölümüne neden oldu.

Kaçan araç sürücüsünün yakalanmasına çalşıldığı bildirildi.
(DA-CÜN-SRP)
07:51 04/01/00

Figure 4.1: The Original News Report

the stemming. The most frequently occuring words and some categorical keywords in the dataset are selected for inclusion in the wild card list Table 4.2.

One of the basic aspect of Turkish phonology is consonant harmony. In one respect, consonants in Turkish may be divided into two groups as voiceless (Ç, F, T, H, S, K, P, S) and the remaining voiced consonants. Turkish words mostly end with a voiceless consonant; especially, the voiced consonants B,C,D or G are rarely found as the final phonemes of the originally Turkish words. If

1 7 78 | ankara okullara kar tatili | ankara kar yağışı okulların tatil edildiği ankara valiliği açıklamada ankara gündür etkili kar yağışı merkez ilçelerinde ilköğretim lise dengi okulların tatil edildiği
1 19 23 | trafik kazası ölü | adana meydana gelen trafik kazasında kişi öldü sürücüsünün kimliği plakası belirlenemeyen bir araç ziyapaşa bulvarı yolun karşısına geçmek isteyen şükrü bulan çarparak ölümüne neden oldu kaçan araç sürücüsünün yakalanmasına çalşıldığı bildirildi.

Figure 4.2: The Preprocessed News Report

ALDA*	BİTT*	DÖNM*	GÖRÜ*
ALMA*	BULAC*	GİD*	KURTULM*
ALMIŞ*	BULM*	GİT*	KURTULA*
ALSIN*	BULD*	GİRE*	KURTULD*
CEZA*	TERÖR*	EĞİT*	JEO*
CENAZE*	DEPREM*	FİLM*	İHALE*
DAVA*	DEVLET*	FUTBOL*	KURUM*
DENİZ*	DUYURU*	FRANS*	TRAFİ*

Table 4.2: Wild Card List

there is one of these consonants at the end of a loan-word, it changes to a corresponding voiceless sound of P, Ç, T and K respectively. Therefore, if a root matching with other words, having different meaning than the word affixed to, exists, both of the words, ending with the voiceless and voiced consonants, are selected as a wild card.

e.g., İLAC and İLAÇ

If **İLA*** is selected as a wild card form in order to comprise both of the words and their extensions, then İLA also contains the words, having a different meaning, such as İLAN, İLAVE, İLAH, İLAHE, İLAHİYAT, İLAHİ

e.g., AĞAC and AĞAÇ

AĞA* is selected as a wild card to contain both of the nouns and their extensions, then AĞA also contain the words such as AĞA, AĞABEY, AĞALIK, AĞANSOY and AĞAR.

GURUP, GURUB, ARAP, ARAB, ARAC, ARAÇ, BORC, BORÇ, HESAP, HESAB are sample words in the aspect of consonant harmony. Therefore, in order to prevent wrongly reduction of the words, both of the words are selected separately as the wild card form, such as GURUP*, GURUB*, ARAP*, ARAB*, ARAC*, ARAÇ*, BORC*, BORÇ*, HESAP*, HESAB*, İLAC*, İLAÇ*, AĞAC* and AĞAÇ*.

However, some words, having consonant harmony aspect, only matches with related words or the words in the same class. Then, both type of the word may be selected in a single wild card.

e.g., KİTAP and KİTABI match the wild card KİTA*.

e.g., KULÜP and KULÜBÜ match the wild card KULÜ*.

e.g., DERNEK and DERNEĞİ match the wild card DERNE*.

ÇOCUK	ÇOCUG̃UN	ÇOCU*
GALİP	GALİBİYET	GALİ*
MENSUP	MENSUBU	MENSU*
BIÇAK	BIÇAG̃IN	BIÇA*
CESET	CESEDİN	CESE*
MEKTUP	MEKTUBA	MEKTU*
MESLEK	MESLEG̃İ	MESLE*
KÖPEK	KÖPEG̃İN	KÖPE*

Table 4.3: Wild Card Form of Softened Voiceless Consonants

HACİM	HACMİNDE	HACMİ*	
AG̃IZ	AG̃ZINDA	AG̃ZI*	
KAYIP	KAYBOLDU	KAYIP*	KAYB*
VAKIF	VAKFI	VAKIF*	VAKF*

Table 4.4: Wild Card Form of Dropped Vowels

e.g., KLİNİK and KLİNİG̃İ match the wild card KLİNİ*.

In multi-syllabic words and in certain mono-syllabic roots, the final voiceless consonant **P, Ç, T, K** are mostly (not always) softened, (it changes **B, C, D or G̃** respectively.) when a suffix, beginning with a vowel, is attached Table 4.3.

Normally, Turkish roots are not flexed. Hovewer, there are some cases where some phonemes are changed by assimilation or various other deformations. These are individual cases and can be treated as exceptions. A root deformation occurs as a vowel ellipsis. When a suffix, beginning with a vowel comes after some nouns,which has a vowel **(I)** or **(İ)** in its last syllable, this vowel drops. In that case, either two forms of the word or the most frequently occuring one is selected as a wild card Table 4.4.

Turkish suffixed can be classified as derivational and inflexional. Derivational suffixes are the suffixes which produce a new word having a different meaning than the word they are affixed to. Inflexional suffixes can be affixed to all of the roots in the class that they belong to.

Derivational suffixes are the most difficult part of the wild card procedure.

Since most of the derivational suffixes change the meaning and the class of the word they are affixed to. Thus, they make nouns from verbs, or verbs from nouns. Therefore, derivational suffixes make the thin line between the stemming and the wild card.
e.g.; EV, EVDE, EVLER, EVDEN, EVLERİMİZİN, EVDEKİLER, have the same root such as (EV) and by removing inflexional suffixes, it is easy for the stemming procedure. However, EVLİ, EVLENMEK, EVLENDİRMEK have the same root but, have the different meaning because of the derivational suffixes. EV* is not a correct decision since many words having a different meaning such as EVLİ, EVLENMEK, EVLENDİRMEK, EVRAK, EVLAT, EVLİYA, EVCİL, EVRİM, EVREN will be in a wrong reduction form by disappearing or reflecting an incorrect categorical meaning.
Then, the solution is not stemming. In order to provide efficiency, each word is a separate wild card as EVİ*, EVLEN*, EVCİL, EVLER*.
e.g.; İŞLEMLERİ, İŞLEDİĞİ, İŞLENEN, İŞLETMESİ have the same root (İŞLE), but various derivational suffixes give different meanings to the each word which are the keywords of different categories. Therefore, each word is formed by a different wild card such as İŞLEM*, İŞLED*, İŞLEN*, İŞLET*.
e.g.; The word (CAN) is a noun, however, many words can be derived from the word by the means of derivational suffixes such as CAN, CANLI, CANKURTARAN, CANLANMAK, CANDAN, CANAN, CANİ,CANLANDIRMAK.
And also root word (PAZAR) can not be in a wild card form, since the derived words such as PAZARLAMA, PAZARCI, PAZARLANDI, PAZARLIK, PAZARBAŞI, are the keywords of different categories.

The wild card form of the words which are affixed by a inflexional suffix, is much easier than the derived words. Since inflexional suffixes can be affixed to all of the roots in the class they belong to and does not change the class of the word they are affixed to.
e.g.; BORSA, BORSADA, BORSADAN, BORSALAR, BORSALARIN, BORSALARIMIZIN have the same root and the wild card BORSA* does not match a word having a different meaning. But, sometimes, in spite of its inflexional suffixes, getting a word to a wild card form is not available.
e.g.; KAMP, KAMPIN, KAMPTA, KAMPLAR, KAMPINDA, KAMPA have

AMA	DOKUZ	EPEY*	PEK
FAKAT	BİN	BİRÇO*	TEKR*
GİBİ	MART	BÖYLE*	MİLYON*
LAZIM	SALI	DÖRT*	MİLYAR

Table 4.5: An Example for Stopwords

the same root (KAMP) and the same meaning. However, KAMP* also comprises the words having a different meaning KAMPANA, KAMPİNG,KAMPANYA, KAMPÜS. Then it is not available to select the word KAMP as a wild card (KAMP*). And also BURS, BURSLU, BURSLAR, BURSLARIN have the same root (BURS) and have tha same meaning but, BURS* also comprises the word BURSA.

The morphome of some words in Turkish prevent automatically removing of inflexional suffixes from the root. e.g., (-LER) and (-LAR) are inflexional suffixes giving a plural meaning to the root. However, the words as DOLAR, KİLER does not allow the removing of plural inflexional suffixes automatically.

In order to provide reduction and retrieve the information, literally matching terms procedure is used. The deficiency of the method is that because most terms have multiple meanings, many unrelated documents may be included in the answer set just because they matched some of the query terms. Against this deficiency, a dimensionality reduction algorithm may be used [16].

The use of a wild card process and deciding which words are eliminated with a stopword list or gathered in a wild card word with a keyword list, can bring about substantial reduction in the number of word variants regarding to Turkish dataset.

Stopwords are words which occur so frequently that they are not useful for distinguishing one document from another, since they are not useful for learning, they are removed Table 4.5.

The most frequently occuring words are mainly function words such as conjunctions, postpositions, pronouns, etc, and those words are selected for

CEZA*	TERÖR*	EĞİT*	JEO*
CENAZE*	DEPREM*	FİLM*	İHALE*
DAVA*	DEVLET*	FUTBOL*	KURUM*
DENİZ*	DUYURU*	FRANS*	TRAFİ*

Table 4.6: An Example for Keywords

BEN	BU	NE
BANA	BUNA	ŞU
SİZ	KİMDEN	BİRİ
ŞUNA	ŞUNLARI	KİME

Table 4.7: Non-Wild Card Pronoun Stopwords

inclusion in the stopword list. Moreover, some of the large number of low-frequency Turkish words are morphological variants of very commonly occuring function words; and these former words are also included in the stopword list.

Keywords are words which are the most commonly used and indicative for a category. Since they are distinguishable from one document to another and provide a reduction for Turkish variants of words, they are selected for inclusion in the keyword list. Keywords may be in the form of wild card Table 4.6.

The stopword and keyword list are created manually and both comprise wild card words, in addition non-wild card words exist in stopword list.

Normally, Turkish roots are not flexed. However, there are some cases where some phonemes are changed by assimilation or various other deformations. An exceptional case related to the flexition of roots is observed in personal pronouns BEN (I) and SEN having datives BANA and SANA respectively. Because of deformations, it is a requirement to select separately most commonly used personal pronouns, demonstrative and relative pronouns as a non-wild card stopword Table 4.7.

Beside the pronouns, not only the name of the days (PAZARTESİ, SALI,...) but also the months (MART, NİSAN,...) are removed as a non-wild card in the stopword list. However, numeral words, most common used Turkish proper

Kİ	DEK	AHMET	ÖZETLE
ÇÜNKÜ	BİLE	AHMED	TARAFINDAN
SANKİ	EĞER	HASAN	ÖTESİNDE
FAKAT	RAĞMEN	ANCAK	MÜTEAKİBEN

Table 4.8: Stopwords without wild cards

names, some of the adverb clauses and prepositions are included in the stopword list as a non-wild card form Table 4.8.

There are four voices of verbs in Turkish: reflexive, reciprocal, causative and passive. Neither the reflexive nor the reciprocal are productive roots; thus, they can be considered as derivational suffixes. But also causative and passive forms sometimes can perform as a derivational one.
e.g.; DÖVMEK, DÖVÜNMEK, DÖVÜLMEK, DÖVÜŞMEK, DÖVÜŞTÜRMEK.
e.g.; GÖRMEK, GÖRÜNMEK, GÖRÜLMEK, GÖRÜŞMEK, GÖRÜŞTÜRMEK.
e.g.; SEVMEK, SEVİNMEK, SEVİLMEK, SEVİŞMEK, SEVİŞTİRMEK.
There are two suffixes which give a verb a negative sense: -M(A) and [Y](A)M(A). And the suffix [Y](A)M(A) is used to express impossibility: YAPMAM, YAPAMAM; OYNAMAM, OYNAYAMAM; SÖYLEMEM, SÖYLEYEMEM. However, the most important point is to decide which form of the verb is a stopword or a keyword. e.g.; The words (ULAŞMAK, ULAŞILMAK, ULAŞTIRILMAK, ULAŞAN, ULAŞAMAYAN, ULAŞMAYAN) have the same root (ULAŞ) and all of them are stopword, but if the verb is in a wild card form (ULAŞ*), some of the keywords (ULAŞTIRMA, ULAŞIM) are discarded from the dataset. Therefore, each form of a verb must be selected one by one to the keyword list (ULAŞTIRMA*, ULAŞIM*) or stopword list (ULAŞIL*, ULAŞTIRIL*, ULAŞA*, ULAŞM*) as a wild card.
The words (DUYDUKLARI, DUYMADIKLARI, DUYGULU, DUYMALARI, DUYULDUĞU, DUYURULAR, DUYULMAMALI) have the same root (DUY), but only DUYURULAR is a keyword of a category, therefore, instead of selecting the root (DUY*) as a wild card, each of the words is taken in the stoplist separately such as DUYD*, DUYG*, DUYM*, DUYUL*, and in the keyword list as DUYURU*.
SATAN, SATACAĞIZ, SATAMAM, SATMAMALI, SATMALI, SATTIRMALI, SATILAN, SATILMAYAN, SATIŞ have the same root (SAT), but most of them

ALDA*	BİTT*	DÖNM*	GÖRÜ*
ALMA*	BULAC*	GİD*	KURTULM*
ALMIŞ*	BULM*	GİT*	KURTULA*
ALSIN*	BULD*	GİRE*	KURTULD*
ALDI*	BULAN*	GİRD*	SÖYLE*
AÇA*	ÇEKTİ*	GİRM*	TANIM*
AÇM*	ÇEKM*	GÖRD*	TANIN*
AÇTI*	ÇEKE*	GİRİL*	TANIT*
AÇI*	ÇEKİN*	GEÇ*	TAŞID*
AŞM*	DURA*	GELE*	TAŞIN*
AŞT*	DURDU*	GELDİ*	TAŞIY*
AŞIN*	DURM*	GÖREN*	TUTM*
AŞILAC*	DÖND*	GÖRM*	TUTT*
BİTİR*	DÖNE*	GÖNDER*	TUTUL*
BİTM*	DÖNÜŞ*	GÖSTER*	UNUT*

Table 4.9: Some Sample Stopword Verbs

are irrelevant words except SATIŞ which is a keyword for a category. Therefore, SATA*, SATM*, SATT*, SATIL* are in the stopword list, however, SATIŞ* is in the keyword list. Some example verbs for stopword list and keyword list exist in Table 4.9 and Table 4.10, respectively.

Main tense suffix is the obligatory suffixes for the verbs. There are nine tenses: definite past (-[D][I]), narrative past (-M[I]Ş), future (-[Y][A]CA[K]), aorist (-[I]R, -[A]R), -R), progressive (-[I]YOR, -M[A]KT[A]), conditional (-S[A]), optative (-[Y][A]), necessative (-M[A]L[I]), and imperative. The last four are not tenses in strict sense of the term, but their place in the verb model is the same as main tense suffix [26]. In Turkish, verb sentences can be transformed into a noun, adjective, or adverb clauses by adding certain suffixes to the verb of the sentence. During the transformation, the obligatory suffixes of the verb, i.e., main tense and personal suffixes, are removed, and then participle suffixes are affixed. Selection process of words by stripping tense and participle suffixes, according to their importance on categories and dataset, is a time consuming task. However, usually, the behavior of tense and participle suffixes reseMbles inflexional suffixes. ÇATIŞMAK, ÇATIŞIR, ÇATIŞTI, ÇATIŞIYOR, ÇATIŞMIŞ, ÇATIŞACAK, ÇATIŞSA, ÇATIŞMALI, ÇATIŞMAYA, ÇATIŞAN, ÇATIŞMAYAN, ÇATIŞMAMAK, have the same

ARTI*	ÇEKİLE*	ONAY*	YARAL*
ANIL*	ÇEKİLİŞ*	ONAR*	YENİL*
ANLAŞ*	ÇEKİM*	ÖDED*	YAKIL*
ARAN*	ÇEKİLM*	ÖDEME*	YANM*
ARAMA*	EĞİT*	ÖDEN*	YANIY*
ARTA*	KAPA*	ÖDEY*	YEND*
ARTMA*	KATIL*	ÖĞRE*	YENE*
ARTT*	KAYB*	SEÇME*	YÜKSELD*
ATAD*	KAÇA*	SEÇİL*	YÜKSELE*
ATAMA*	KAÇM*	SEÇTİ*	YÜKSELİ*
AVLANM*	KAÇT*	PATLA*	YÜKSELT*
BİRLEŞM*	KAÇIR*	SALDIR*	YÜKSELME*
BİRLEŞT*	KALKIN*	TUTUK*	YIKT*
ÇEKİLD*	OYNA*	VURUL*	YIKIL*

Table 4.10: Some Sample Keyword Verbs After Stopword Elimination

wild card ÇATIŞ*. In some cases, the main tense and participles suffixes causes deformation on some stems they are affixed to. e.g., İSTEMEK, İSTEDİ, İSTER, İSTİYOR.
Because of deformations and matching with other words, it is sometimes not available to gather all tense forms of a verb in a wild card. e.g.,BAKMAK, BAKAR, BAKIYOR, BAKACAK, BAKTI, BAKSA,BAKMIŞ, BAKMIŞTI, BAKMADI, BAKILDI, BAKILIYOR, BAKILSA, BAKILIR, BAKILMIŞ, have the same root (BAK), but the wild card (BAK*) also comprise the words, (BAKAN, BAKANLIK, BAKIM, BAKIR) which have different meanings. Therefore, each tense form of the verb is taken separately in the stopword list such as BAKMA*, BAKT*, BAKIL, BAKIY*, and in the keyword list such as BAKAN*, BAKIM*. Different forms of verbs in the stoplist are in Table 4.11.

A keyword is the most frequently occuring indicative word for a *category*, but a stopword is the most frequently occuring word for the *whole dataset*. And also the number of *verbs* in the stopword list is more than the number of *nouns*, but, in contrast, the number of *nouns* in the keyword list is more than the number of *verbs*. In fact, each category has several keywords that indicates the category Table 4.12.

BAKT*	BİTT*	DUYURD*	OLAS*
BAKMA*	BULAC*	DUYURM*	OLDU*
BAKIL*	BULM*	DUYUY*	OLMUŞ*
BAKIY*	BULD*	İSTE*	OLMA*
BİLDİ*	BULAN*	İTİY*	OLU*
BİLGİL*	BULU*	İNANC*	OLACA*
BİLİN*	DEMEK*	İNANI*	YAPIY*
BİLİY*	DEDİ*	İNAND*	YAPIL*
BİLM*	DEMİŞ*	İNANM*	YAPM*
BİLDİR*	DENİL*	OLAB*	YAPTI*
BİTİR*	DİYE*	OLAM*	YAPA*
BİTM*	DİYO*	OLAN*	YÜRÜY*

Table 4.11: Some Sample Tense Forms of Stopword Verbs

CATEGORY	KEYWORDS			
EĞİTİM	eğit*,	öğre*,	okul*,	bursl*.
SEÇİM	seç*,	oyl*,	aday*,	oy.
BORSA	borsa*,	bono*,	imkb,	hisse.
İHALE	ihale*,	duyuru*,	sözleşme*,	fesh*.
DİN	peygamber,	haç,	dinci,	müslüman*.
HÜKÜMET	hükümet*,	başbakan*,	bakan*,	ecevit.
DEPREM	deprem*,	sarsı*,	artçı,	richter.
YANGIN	itfaiye*,	yangın*,	yanm*,	söndür*.
HUKUK	mahk* ,	anayasa,	yarg*,	yasas*.
TERÖR	terör,	bomba*,	örgüt*,	pkk.
ÇATIŞMA	çatış*,	çeçen*,	mevzi*,	saldır*.
DÖVİZ	döviz*,	efektif,	kur,	alış.
PARTI	politik*,	parti*,	siyas*,	muhalefet*.
MECLIS	milletvek*,	delege*,	tbmm,	parlamen*.

Table 4.12: Some Sample Keywords of Categories

4.4 Categories

The most important and time consuming task of the preprocessing is to assign categories to documents. Since the original AA dataset is unprocessed that is the categories are not assigned, the documents are manually indexed using 78 subject categories. Each category is represented by a number, related to a subject Table 4.13.

In general, the structure of categories is flat, but a category hierarchy is requirement in order to provide an efficient categorization Table 4.13. Most of the categories are organized in a hierarchy. For example; C_1 : Sports, C_2 : Football, C_3 : Basketball, C_4 : Volleyball, and C_5 : Athletics. Needless to say that all categories, football, basketball, volleyball and athletics are a branch of sports. However, some news reports require to be assigned the category *sports* except for the branches of sports. Then such news are assigned only the category *sports*. On the other hand, if a news report belongs to a particular category in the hierarchy, such as C_2 : Football, since Football is the branch of Sports, the news report is also assigned to C_1 : Sports.

Category boundaries are defined by whitespace and categorization numbers are assigned sorted. On average 7 categories are assigned to each document. That the way followed while manual categorizing is to research both the general subject of the document and the most commonly used keywords in the document.

4.5 Feature Values

In text categorization, instances are natural language documents and features or terms are the words. An instance can be represented with a set of word and word weight pairs. Each feature has a weight in a specific instance and this weight indicates the importance of that feature in the instance. Finding out the proper weight of features is a crucial process in an algorithm since it highly effects the accuracy of the algorithm. A succesful feature weighting technique gives low weight to features of which almost every instance contains nearly the

1. İÇ	5. TERÖR	9. SANAT	13. MAGAZİN
2. DIŞ	6. ÇATIŞMA	10. VEFAT	14. TURİZM
3. HUKUK	7. EĞİTİM	11. ANMA	15. DOĞA
4. ASAYIŞ	8. KÜLTÜR	12. BİLİM	16. SAĞLIK
17. TARIM	21. HAVAYOLLARI	25. BM	29. PETROL
18. HAYVANLAR	22. DEMIRYOLLARI	26. ORDU	30. ORTADOGU
19. TRAFIK	23. KAZA	27. IHALE	31. BARAJ
20. DENIZ	24. AB	28. DIN	32. IÇME SUYU
33. SEÇIM	34. INSAN HAKLARI	35. SANAYİ	
36. SPOR			
37. FUTBOL	38. BASKETBOL	39. VOLEYBOL	40. HENTB0L
41. ATKETIZM	42. KAYAK	43. SU SPORLARI	44. MINDER
45. BILARDO	46. RALLI	47. AT YARISI	48. OLIMPIYAT
49. EKONOMI			
50. BORSA	51. DÖVIZ	52. IHRACAT	53. ITHALAT
54. ZAM	55. ENFLASYON	56. BANKACILIK	57. TICARET
58. IMF	59. HAZINE		
60. YÖNETIM			
61. DEVLET	62. HÜKÜMET	63. PARTILER	64. BELEDIYE
65. PARLAMENTO	66. SIVIL TOPLUM	67. KURULUSLAR	68. KONSEY
69. AFET			
70. YANGIN	71. DEPREM	72. SEL	73. FIRTINA
74. AÇLIK	75. SALGIN	76. YANARDAG	
77.TALIH OYUNLARI	78. METEOROLOJI		

Table 4.13: Categories

same amount. Since they are irrelevant to the categorization of the instances. Because of the importance of feature weighting process, a growing number of techniques have been developed and applied, e.g., term frequency (TF), inverse document frequency (IDF), term discrimination value, probabilistic term weighting, single feature accuracy and genetic algorithms.

Term frequency is the number of occurences of a term in which a document occurs. In short, the more often a term occurs in a document, the more likely it is to be important for that document. However, because of the fact that, the number of occurences of a term depends on the length of the instance, in general, it is normalized by the number of the terms in the instance.

As mentioned in Section 4.2, each news report is divided into three bodies. Headline text body is a short summary of the instance and in general, contains relevant features regarding to the instance. Therefore, while computing term frequency, all features in the text body have the same weight coefficient, in contrast, the features in the headline body have higher weight coefficient than the ones in the text body. The aim is to retrieve more relevant features in the instance.

TRAFİK KAZASI: 1 ÖLÜ...

ADANA (A.A) - Adana'da meydana gelen trafik kazasında bir kişi öldü.
Alınan bilgiye göre, sürücünün kimliği ve plakası belirlenemeyen bir araç, Ziyapaşa Bulvarı'nda yolun karşısına geçmek isteyen Şükrü Bulan'a (80) çarparak, ölümüne neden oldu.
Kaçan araç sürücüsünün yakalanmasına çalşıldığı bildirildi.

Figure 4.3: A Sample Instance

The headline body of Fig 4.3 is related with a traffic accident and 1 person is dead. And the features in the headline are keywords, however, the text body implies the same subject in detail and also contains many irrelevant features. Therefore, in order to retrieve more relevant features, assigning a higher weight coefficient to the features in the headline is so reasonable Fig 4.4.

1 19 23 | adana* 1 araç* 2 bulv* 1 kaza* 4 kaça* 1 kimliğ* 1 plaka* 1 sürücü* 2 trafi* 4 yakala* 1 yol* 1 ziyapaşa 1 çarp* 1 öl* 5

Figure 4.4: Term Frequency of an Instance

Chapter 5

Evaluation

This chapter is devoted to the empirical evaluations of the FPTC and k-NN algorithms on the real dataset. Anadolu Agency Newsgroup Dataset is used to compare the predictive power and computational complexity of those algorithms.

The k-NN classifier is an instance based learning method. It computes the similarity between the test instance and training instance, and considering the k top-ranking nearest instances to predict the categories of the input, finds out the category that is most similar. The FPTC is a variant of k-NN algorithm [12]. The main difference is that instances are projected on their features in the n dimensional space and distance between two instances is calculated according to a single feature. The Feature Based Text Categorization (FBTC) algorithm is an extension of the FPTC. The whole dataset is viewed as a matrix where each row is an instance and its columns are features. The values in the matrix are binary, 1 for presence of the feature and 0 for absence of the feature. In other words, the main difference is the feature weighting mechanism. Instead of taking term frequency of a feature, if the feature occurs in the instance, 1 is assigned as the weight of the feature, otherwise 0 is assigned.

The organization of the chapter is as follows: the performance measure that will be used to compare the methods in terms of predictive accuracy is briefly defined. Then, computational complexity comparison of the algorithms

in terms of time and space requirements is presented. In the third section, empirical evaluation including the aspects of the real dataset used in experiments, is mentioned. Finally, the chapter concludes by the evaluation results and the comparisons of algorithms.

5.1 Performance Measure

The performance measure is used to determine the predictive power of the algorithms and several measures for performance exist. One of the performance measures of a categorization algorithm is to find its accuracy. The most commonly used categorization accuracy metric is the percentage of correctly categorized instances over all test instances for a given dataset. As mentioned in *Chapter 2*, the k-NN and FPTC algorithms use category ranking to predict the categories of the input document. Category ranking can be evaluated using measures similar to the conventional measures for evaluating ranking-based document retrieval systems: recall, precision, and 11-point average precision. Given a categorizer whose input is a document, and whose output is a ranked list of categories assigned to that document, the recall and precision can be computed at any threshold on this ranked list:

$$recall = \frac{categories\ found\ and\ correct}{total\ categories\ correct} \tag{5.1}$$

$$precision = \frac{categories\ found\ and\ correct}{total\ categories\ found} \tag{5.2}$$

where "categories found" specifies some number of the categories from the beginning of the ranked list. This number can adjust according to the algorithms. In the k-NN algorithm, the relevant number is the parameter k and in the FPTC, it is the number of all categories. "Categories found and correct" means how many categories from the found categories agree with the actual categories given to the tested instance manually. "Total categories found " is the number of the total categories found and "Total categories correct" is the number of actual categories of the tested instance.

After finding recall and precision for each test instance, the global evaluation of an algorithm can be found by interpolated 11-point average precision that is described below [38]:

- As the first step, for each document, compute the recall and precision at each position in the ranked list where a correct category is found.
- For each interval of recall values 0%, 10% ...,100% the left boundary of the interval is assigned to the highest precision value of that interval. For example, the highest precision value in the interval of 0% and 10% recall values is assigned to the recall value of 0%. If the interval value contains no precision value, then zero is assigned as the precision value to the left boundary of that interval. These precision values of recall values are called as representative precision value.
- For the recall value of 100% representative precision is either the exact precision value if such a data point exists, or the precision value at the closest point in terms of recall.
- The representation value of each recall values of 0%...100% is replaced with the highest representative precision value of that recall value or higher recall values. This process is called as *interpolation*.
- Average per-document data points over all the test documents, at each of the above recall thresholds respectively. This step results in 11 per-interval average precision scores.
- The average of these per-interval average precision scores specifies a single-numbered performance average, called global averaging.

Accuracy of an algorithm is the measure of correct categorizations on the test set of unseen instances. In order to find accuracy, while applying precision, recall and 11-point average precision, 10-fold cross-validation technique is used. The whole dataset is partitioned into 10 subsets. The nine of the subsets is used as the training set and the tenth is used as the test set, and this process is repeated 10 times once for each subset being the test set. Therefore, each instance appears once in the test set and nine times in the training and categorization is the average of these 10 runs.

5.2 Complexity Analyses

Since all the training instances are stored in the memory in both k-NN and FPTC algorithms, the space required for training with m instances on a domain with n features is proportional to m n. That is, the space complexities of these algorithms are $O(m \times n)$.

In the training, all instances are stored on each feature dimension as their feature projections in the FPTC algorithm. And then they are sorted once at the end. Since the sorting of m feature values has the time complexity of $O(m \times logm)$. For a dataset containing m instances and n features, the training time complexity of the FPTC is $O(n \times m \times logm)$. On the other hand, the k-NN algorithm has the time complexity of $O(m \times n)$ for storing all instances in memory.

The $kBag(f, t, k)$ function, to determine the votes of a feature, first finds the nearest neighbor of t on f and then next $k-1$ neighbors around the nearest neighbor. The time complexity of this process is $\mathbf{O}(log\ m + k)$. Since $m >> k$, the time complexity of $kBag$ is $\mathbf{O}(log\ m)$. The final classification requires the votes of each of n features. Therefore, the classification time complexity of the FPTC algorithm is $\mathbf{O}(n \times (k + log\ m))$ [11].

In the **k-NN** algorithm, the classification of a test instance requires the computation of its distance to m training instance on n dimensions. Therefore, the classification time complexity of the k-NN algorithm is simply $\mathbf{O(n \times m)}$ assuming $m >> k$.

5.3 Empirical Evaluation

In this section, we present an empirical evaluation of the FPTC algorithm on the A.A. dataset and the results will be compared with that of the k-NN algorithm.

5.3.1 Real-World Dataset

The FPTC and k-NN algorithms are evaluated on the A.A. real-world dataset. The processed dataset consists of nearly 2000 documents. A stopword list of 859 entries is used for stopword elimination. Each news report contains a categorized number body, a headline text and news text body. The headlines are an average of 12 words long. The average length of a document body is 96 words. On average 7 categories are assigned to each document. Text categorization is such an application area where datasets inevitably contain a high number of irrelevant features. Even after removal of stop words, many irrelevant features still exist and also there are many "noisy" data which makes the categorization difficult to learn for a categorizer. The original A.A. (Anadolu Agency) dataset is unprocessed, that is, the categories were manually assigned to subjects using 78 subject categories. Each category label is represented by a number defined a subject. Word boundaries were defined by whitespace. Some preliminary work was required besides converting all words to lower case and removing punctuation marks because of the characteristics of Turkish language.

5.3.2 Experimental Results

An empirical evaluation of the FPTC algorithm is presented in order to show the evaluation and the comparison the performance of the FPTC with the k-NN categorizer algorithm on the A.A. real-world dataset. The parameter "k" is defined as 10 in the experiments. The experiments also gives the opportunity of

- verifying the previously obtained results about the comparison of the FPTC and the k-NN algorithms in text categorization [12].
- observing the performance of the algorithms on the Turkish language.

The accuracy results of the FPTC and FBTC algorithms for each cross-validation are given in Table 5.1 and Table 5.2, respectively. We can say that the FPTC algorithm gives promosing results in the text categorization of the Turkish real-world dataset.

Fold	Accuracy	Training Time (ms)	Test Time (ms)
1	0.863	439180	18460
2	0.793	441600	17630
3	0.845	444450	17470
4	0.829	446650	17090
5	0.855	444780	18070
6	0.791	433690	17410
7	0.839	442700	18400
8	0.899	442480	17860
9	0.844	438090	17410
10	0.964	447150	17740
Avg	0.8612	442077	17754

Table 5.1: The Results of FPTC for each cross-validation

Fold	Accuracy	Training Time (ms)	Test Time (ms)
1	0.729	429300	17740
2	0.587	432540	18400
3	0.748	431440	17080
4	0.691	430840	18180
5	0.68	431440	18180
6	0.64	428590	17240
7	0.661	426770	18510
8	0.741	430560	18510
9	0.719	424030	18510
10	0.946	430390	17960
Avg	0.7142	429590	18031

Table 5.2: The Results of FBTC for each cross-validation

Algorithm	Accuracy	Training Time (ms)	Test Time (ms)
FPTC	0.863	439180	18460
FBTC	0.729	429300	17740
k-NN	0.941	870920	63341280

Table 5.3: The Comparison of the Algorithms after the first fold cross-validation

According to the results in Table 5.3, the k-NN is slightly better than the FPTC algorithm in terms of accuracy, the difference between their accuracy is nearly 7%. However, the FPTC and the FBTC are much better than the k-NN algorithm in terms of time. During the testing of the instances, one fold cross-validation of the FPTC algorithm lasts only 18 seconds,whereas one fold cross-validation of k-NN lasts nearly 18 hours Table 5.3. According to the comparison of time efficiencies of the algorithms, the accuracy difference among the algorithms can be negligible.

In short, after the evaluation of the algorithms, the results show that the accuracy of the FPTC algorithm is so close to the accuracy of the k-NN algorithm, whereas the time efficiency of FPTC outperforms the k-NN significantly.

Chapter 6

Conclusion and Future Work

In this thesis, compilation and preprocessing of a new Turkish dataset, and also evaluation and comparison of the FPTC and the k-NN text categorization algorithms on the Turkish dataset are presented.

Unlike the main Indo-European languages, such as French, English and German, Turkish is an example of an agglutinative language, where words are formed by affixing morphemes to a root in order to extend its meaning or to create other classes of words. In Turkish the process of adding one suffix to another can result in relatively long words, which often contain an amount of semantic information equivalent to a whole English phrase, clause or sentence. Due to this complex morphological structure, Turkish requires text processing techniques different than English and similar languages. However, the Turkish real-world data collection on which the FPTC and the k-NN algorithms are applied, achieves comparable performance with the English or similar language data collections.

The FPTC algorithm has been compared with the k-NN algorithm in terms of categorization accuracy and time complexity on the Turkish dataset. The k-NN algorithm is a well-known text categorization algorithm and comparable good in accuracy, but not in scalability. The FPTC algorithm is a variant of k-NN and a non-incremental algorithm that is all training instances are taken and processed at once. The main characteristic of the algorithm is that

instances are stored as their projections on each feature dimension. If the value of a training instance is missing for a feature, that instance is not stored on that feature.

On the Turkish real-world dataset, the FPTC algorithm achieves comparable accuracy with the k-NN algorithm. On the other hand, the average running time of of the FPTC algorithm is much less than that of the k-NN algorithm.

As a future work, we plan to increase the number of the processed Turkish news reports in the dataset. And also a standard stopword list can be developed for the Turkish language. Another research direction is to increase the prediction accuracy of the FPTC algorithm on the Turkish dataset. And the effect of the missing and noisy feature values on the prediction accuracy of the FPTC algorithm can be investigated as a future work.

Bibliography

[1] Apte, C., Damerau, F., Weiss, S., Towards Language Independent Automated Learning Of Text Categorization models, *In Proceedings of the* 17^{th} *Annual ACM/SIGIR Conference*, 1994.

[2] Aha D. W., Mohri T., Dietrich W., Turkic Language Automation: Existing Software and Turkic Language Automation, Samarkand, 1995.

[3] Arzikulov, H., Turkic Language Automation: Existing Software and Turkic Language Automation, Samarkand, 1995.

[4] Cohen, W. W., and Singer, Yoram., D., Kibler, D., Context-sensitive Learning Methods for Text Categorization, *In the Proceedings of the* 19^{th} *Annual ACM/SIGIR Conference on Research and Development in Information Retrieval*, 1996.

[5] Cohen, W. W., Fast Effective Rule Induction *In Machine Learning: Proceedings of the Twelfth International Conference, ML95*, 1995.

[6] Cohen, W. W., Learning Trees and Rules with Set-Valued Features, *In Proceedings of* 13^{th} *National Conference on Artificial Intelligence*, 1996.

[7] Dasarathy, B. V., Nearest Neighbor(NN) Norms: NN Pattern Classification Techniques, , *IEEE Computer Society Press*, Los Alamitos, 1991.

[8] Ekmekcioglu, F. C., Lynch, M. F. and Willett, P., Stemming and N-gram Matching for Term Conflation In Turkish Texts,

[9] Furnkranz, J., Mitchell, T., and Riloff, E., A Case Study in Using Linguistic Phrases for Text Categorization on the WWW,

[10] Gale, W. A., and Lewis, D. D., A Sequential Algorithm for Training Text Classifiers. *In the Proceedings of the* 17^{th} *International Annual ACM/SIGIR Conference*, 1994.

[11] Güvenir,H. A., and Akkuş, A., k Nearest Neighbor Classification on Feature Projections, In *Proceedings of the* 13^{th} *International Conference on Machine Learning*. Lorenza Saitta (Ed.), Bari, Italy: Morgan Kaufmann, 12–19, 1996.

[12] Güvenir,H. A., and Yavuz, T., Application of k Nearest Neighbor on Feature Projections Classifier to Text Categorization *In Advances in Computer and Information Sciences'99 (Procedings of the* 13^{th} *International Symposium on Computer and Information Sciences-ISCIS'98)*, Oct.26–28,1998.

[13] Güvenir, H. A., Şirin, İ., Classification by Feature Partitioning, *Machine Learning*, Vol.23, No:1, 47-67, 1996.

[14] Han, E., Karypis, G., Kumar, V., Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification University of Minnesota, Minneapolis, USA. *In Proceedings of The Twelfth International Joint Conference on Artificial Intelligence*, 1991.

[15] Hayes, P. J., and Weinstein, S. P., Construe: A System for Content-Based Indexing of a Database of News Stories, *In Proceedings of the Second Annual Conference on Innovative Applications of Intelligence*, 1990.

[16] Karypis, G., and Han, E., Concept Indexing A Fast Dimensionality Algorithm with Applications to Document Retrieval and Categorization University of Minnesota, Minneapolis, USA, 2000.

[17] Kelly, J.D., and Davis, L., A Hybrid Genetic Algorithm for Classification, *In Proceedings of The Twelfth International Joint Conference on Artificial Intelligence*, 1991.

[18] Larkey, L. S., Automatic Essay Grading Using Text Categorization Techniques, *In the Proceedings of the* 17^{th} *International Annual ACM/SIGIR Conference*, 1998.

[19] Liere R., and Tadepalli., The Use of Active Learning in Text Categorization, *Department of Computer Science, Oregon State University*,

[20] Lewis D. D., and Ringuette., A Comparison of Two Learning Algorithms, *In Proceedings of The Third Annual Symposium on Document Analysis and Information Retrieval*, 1994.

[21] Lewis D. D., Schapire, R. E., Callan., J. P., and Papka, R., Training Algorithm For Linear Text Classifiers, *In the Proceedings of the* 19^{th} *International Annual ACM/SIGIR Conference*, 1994.

[22] Lovins, J. B., Development of a Stemming Mechanical Algorithm, *Mechanical Translation and Computational Linguistics*, 1968.

[23] Mitchell, T.M., Machine Learning, *McGraw Hill*, 1997.

[24] Moulinier, I., A Framework for Comparing Text Categorization Approaches, *LAFORIA-IBP-CNRS*, PARIS,1996.

[25] Moulinier, I., Is Learning Bias an Issue on the Text Categorization Problem, *LAFORIA-LIP6,Universite Paris*, 1997.

[26] Oflazer,K., and Solak, A., Design and Implementation of a Spelling Checker, *Master's Thesis, Bilkent University, Dept. of Computer Engineering Science, Ankara*, 1991.

[27] Pederson. J. O., Wiener, E., and Weigned, A. S., A Neural Network Approach to Topic Spotting, *In Proceedings of The Fourth Annual Symposium on Document Analysis and Information Retrieval, (SDAIR'95)*, 1995.

[28] Raskinis, G., Ganascia, J., and Moulinier, I., Text Categorization: A symbolic Approach *In Proceedings of The Fifth Annual Symposium on Document Analysis and Information Retrieval, (SDAIR'96)*, 1996.

[29] Ruiz, M. E., Srinivasan, P., Automatic Text Categorization Using Neural Network *In Proceedings of The 8th ASIS SIG/CR Classification Research Workshop* , New Jersey, 1998.

[30] Salton, G., Automatic Text Processing: The Transformation Analysis and Retrieval of Information by Computer, *Addison-Wesley, Reading, Pennsylvania*, 1989.

[31] Salzberg, S., Distance Metrics for Instance Based Learning, *ISMIS'91* 6^{th} *International Symposium Methodologies for Intelligent Systems*, 1991.

[32] Salzberg, S., Delcher, A., Heath, D., and Kasif, S., Best-Case Results For Nearest Neighbor Learning, *IEEE Transactions on Pattern Analyis and Machine Intelligence*, 17:599-560, 1991.

[33] Scott, Sam., and Marvin, S., Text Classification Using WordNet Hypernyms,

[34] Waltz, D., and Stanfill, C., Toward memory-based reasoning, *Communications of the Association for Computing Machinery*, 29:1213-1228, 1986.

[35] Weiss, S. A., Kasif, S., and Bill, E., Text Classification in USENET Newsgroup: A Progress Report, *In Proceedings of The AAAI Spring Symposium on ML in Information Access*, 1996.

[36] Wettschereck, D., and Dietterich, T. G., An Experimentsl Comparison of the Nearest hyper-rectangle Algorithms, *Machine Learning*, 9:5-28, 1995.

[37] Wilson, D. R., and Martinez, T. R., Improved Heterogeneous Distance Functions, *Journal of Artificial Intelligence Research*, vol:6, no: 1, 1-34, 1997.

[38] Yang, Y., An Evaluation of Statiscal Approaches to Text Categorization, *School of Computer Science, Cornegie Mellon University, Kluwer Academic Publishers*, 1999.

[39] Yang, Y., Sampling Strategies and Learning Efficiency in Text Categorization, *In AAAI Spring Symposium on Machine Learning in Information Access*, 1996.

[40] Yang, Y., Noise Reduction in a Statistical Approach to Text Categorization, *In the Proceedings of the* 18^{th} *International Annual ACM/SIGIR Conference*, 1995.

[41] Yang, Y., and Chute., C. G., An Example-based Mapping Method for Text Categorization and Retrieval, *ACM Transaction on Information Systems (TOIS)*, 1994.

[42] Yang, Y., Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval, *In the Proceedings of the* 17^{th} *International Annual ACM/SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*, 1994.

[43] Yang, Y., and Pederson, J. O., A Comparative Study on Feature Selection in Text Categorization,