EMOTION ANALYSIS OF TURKISH TEXTS BY USING MACHINE LEARNING METHODS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ZEYNEP BOYNUKALIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JULY 2012

Approval of the thesis:

EMOTION ANALYSIS OF TURKISH TEXTS BY USING MACHINE LEARNING METHODS

submitted by **ZEYNEP BOYNUKALIN** in partial fulfillment of the requirements for the degree of
Master of Science in Computer Engineering Department, Middle East Technical University by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _____
Head of Department, **Computer Engineering**

Assoc. Prof. Pınar Şenkul _____
Supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Adnan Yazıcı
Computer Engineering Dept., METU _____

Assoc. Prof. Pınar Şenkul
Computer Engineering Dept., METU _____

Prof. Dr. Ferda Nur Alpaslan
Computer Engineering Dept., METU _____

Asst. Prof. Çiğdem Gündüz Demir
Computer Engineering Dept., Bilkent University _____

Dr. Selçuk Köprü
Teknoloji Yazılımevi _____

Date: _____

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

Name, Last Name: ZEYNEP BOYNUKALIN

Signature :

ABSTRACT

EMOTION ANALYSIS OF TURKISH TEXTS BY USING MACHINE LEARNING METHODS

Boynukalın, Zeynep
M.Sc., Department of Computer Engineering
Supervisor : Assoc. Prof. Pınar Şenkul

July 2012, 69 pages

Automatically analysing the emotion in texts is in increasing interest in today's research fields. The aim is to develop a machine that can detect type of user's emotion from his/her text. Emotion classification of English texts is studied by several researchers and promising results are achieved. In this thesis, an emotion classification study on Turkish texts is introduced. To the best of our knowledge, this is the first study on emotion analysis of Turkish texts. In English there exists some well-defined datasets for the purpose of emotion classification, but we could not find datasets in Turkish suitable for this study. Therefore, another important contribution is the generating a new data set in Turkish for emotion analysis. The dataset is generated by combining two types of sources. Several classification algorithms are applied on the dataset and results are compared. Due to the nature of Turkish language, new features are added to the existing methods to improve the success of the proposed method.

Keywords: Emotion Analysis, Text Classification, Machine Learning, Turkish Text, Text Mining

ÖZ

MAKİNE ÖĞRENİMİ TEKNİKLERİYLE TÜRKÇE METİNLERDE DUYGU ANALİZİ

Boynukalın, Zeynep
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Pınar Şenkul

Temmuz 2012, 69 sayfa

Yazılı metinlerin otomatik analiz edilerek içerdiği duyguyu belirlemek araştırma alanlarında gün geçtikçe daha çok önem kazanmaktadır. Genel amaç kullanıcının yansıttığı duyguyu çıkarabilen bir makine geliştirmektir. İngilizce metinlerde duygu tespiti bazı araştırmacılar tarafından çalışılmış ve ümit veren sonuçlara ulaşılmıştır. Yaptığımız araştırmalarda Türkçe metinlerde duygu analizi üzerinde yapılmış bir çalışma bulunamamış ve bu tez kapsamında bu konu üzerinde çalışılmıştır. İngilizce için iyi tanımlanmış ve konu için uygun veri setleri mevcuttur, ancak Türkçe üzerinde çalışabilmek için böyle bir veri seti bulunamamıştır. Bu tezle beraber, iki farklı kaynaktan yararlanılarak yeni bir veri seti tanımlanmış ve çalışmada bu set kullanılmıştır. Farklı otomatik öğrenme teknikleri denenmiş ve sonuçlar karşılaştırılmıştır. Türkçenin İngilizceden ayrılan özellikleri nedeniyle kullanılan yöntemlere eklemeler yapılarak Türkçe metinlerde yapılan analizin başarısı artırılmaya çalışılmıştır.

Anahtar Kelimeler: Duygu Analizi, Metin Sınıflandırması, Makine Öğrenimi, Tükçe Metin, Metin Madenciliği

To my family, especially my little nephew.

ACKNOWLEDGMENTS

I express my sincere appreciation to my supervisor, Assoc. Prof. Pınar Şenkul, for her guidance, insight, support, encouragement and positive attitudes throughout the study. I am thankful for having the chance of working with her.

I would like to thank all my friends for their helps, supports and valuable efforts on the data collecting process.

I am very grateful and would like to thank my family for their invaluable patience and encouragement, and for being the source of my motivation.

I would like to thank my company ASELSAN Inc. for supporting my thesis.

Finally, I would like to thank TÜBİTAK (The Scientific and Technological Research Council of Turkey) for providing financial support during my thesis.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ v

ACKNOWLEDGMENTS vii

TABLE OF CONTENTS viii

LIST OF TABLES x

LIST OF FIGURES xii

CHAPTERS

1 INTRODUCTION 1

1.1 Emotion in Texts 2

1.2 Turkish Language 3

1.3 Contribution and Organization of Thesis 3

2 LITERATURE SURVEY 5

2.1 Emotion Analysis of Texts 5

2.2 Studies on Analysis of Turkish Texts 8

3 BACKGROUND 10

3.1 Classification Methods 10

3.1.1 Naive Bayesian Classifier 11

3.1.2 Complement Naive Bayes 12

3.1.3 Support Vector Machines 12

3.1.4 Feature Selection and Feature Vector Construction 14

3.1.4.1 N-Grams 14

3.1.4.2 Weighted Log Likelihood Ratio 15

3.1.4.3 Feature Vector Construction 16

3.1.5 WEKA 17

3.2 Natural Language Processing 18
3.2.1 Zemberek 19
4 PROPOSED METHODS 21
4.1 Data Gathering 22
4.1.1 ISEAR Dataset 23
4.1.2 Turkish Fairy Tales Dataset 23
4.1.2.1 Data Annotation Process 25
4.2 Data Preprocessing 30
4.2.1 Stop Word Removal 32
4.2.2 Morphological Analysis 32
4.2.2.1 Negation Handling 33
4.2.2.2 Handling Special Suffixes 34
4.3 Feature Selection and Feature Vector Construction 36
4.3.1 Generating the Feature Scores 36
4.3.2 Feature Vector Construction 37
4.4 Classification 39
5 EXPERIMENTAL RESULTS 42
5.1 Classification Results of ISEAR Dataset 43
5.2 Classification Results of Turkish Fairy Tales Dataset 46
5.2.1 Results With Emotion Classes 47
5.2.2 Results With Emotional Levels 51
5.3 Classification Results of Combined Dataset 52
5.4 Evaluation of Overall System and Discussion 56
5.5 Experiments With Articles 57
6 CONCLUSION AND FUTURE WORK 60
REFERENCES 63
A LISTS 67
A.1 Stop Words List 67
A.2 Fairy Tales List 68
A.3 Most Distinctive Words of Combined DataSet 69

LIST OF TABLES

TABLES

Table 3.1 Example Set of N-grams 15

Table 4.1 Samples from ISEAR Data 24

Table 4.2 Distribution of ISEAR Dataset 25

Table 4.3 Examples from Emotional Words List 25

Table 4.4 Samples from Fairy Tales Data 26

Table 4.5 Agreement Matrix of Two Annotators, J1 and J2 28

Table 4.6 Pairwise Agreement of Emotion Categories 28

Table 4.7 Pairwise Agreement of Emotion Levels 29

Table 4.8 Distribution of Fairy Tales Data 29

Table 4.9 Fairy Tales Emotion Levels 30

Table 4.10 Distribution of Combined Data 30

Table 4.11 Spell Checking with Zemberek 31

Table 4.12 Examples from Stop Words List 32

Table 4.13 Samples for Negation Suffixes of Zemberek 33

Table 4.14 Specially Treated Suffixes 34

Table 4.15 Example of Sentence Preprocessing 35

Table 4.16 N-grams Count for Each Dataset 36

Table 4.17 Most Distinctive Features for ISEAR Dataset 37

Table 4.18 Most Distinctive Features for Fairy Tales Dataset 38

Table 5.1 Classification Results of ISEAR Dataset with NB Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods 43

Table 5.2 Classification Results of ISEAR Dataset with CNB Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods 44
Table 5.3 Classification Results of ISEAR Dataset with SVM Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods 45
Table 5.4 Confusion Matrix of ISEAR Dataset 46
Table 5.5 Classification Results of Fairy Tales Dataset with NB Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods . . . 47
Table 5.6 Classification Results of Fairy Tales Dataset with CNB Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods 48
Table 5.7 Classification Results of Fairy Tales Dataset with SVM Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods 49
Table 5.8 Confusion Matrix of Fairy Tales Dataset 50
Table 5.9 Classification Results of Two Categories with Different Classifiers 50
Table 5.10 Accuracies of Emotion Intensity Levels 51
Table 5.11 Classification Results of Combined Dataset with NB Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods 52
Table 5.12 Classification Results of Combined Dataset with CNB Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods 53
Table 5.13 Classification Results of Combined Dataset with SVM Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods 54
Table 5.14 Confusion Matrix of Combined Dataset 55
Table 5.15 Accuracies of Test Set with Different Classifiers 56
Table 5.16 Standard Deviations of Cross Validation Results 56

LIST OF FIGURES

FIGURES

Figure 3.1 An Example of Separation in 2d (Figure taken from [36]). 13

Figure 3.2 A Screenshot of Weka 17

Figure 3.3 Example Analysis of Zemberek 19

Figure 3.4 Suggestion of Zemberek 19

Figure 4.1 Overview of the System 22

Figure 5.1 Percentage Accuracy versus Feature Set Diagram for Classification with CNB Classifier and Combined Dataset 55

CHAPTER 1

INTRODUCTION

Improvements in technology and ease of using it, made computers to be irreplaceable part of our lives. People being so addicted and connected to computers, revealed the advanced studies of human computer interaction. Human computer interaction (HCI) is a discipline concerned with improvement of interactions between users and computers by making the computers easy to use and more importantly by making computers more effective for our lives. The areas that make use of HCI are wide, ranging from categorizing web contents to analysis of public opinion on a fact or customer feedback. Sentiment analysis and emotion analysis are the two different research areas that are especially important in HCI, as Nass's studies on human-human and human-computer interactions say, people most naturally interact with their computers in a social and affectively meaningful way, just like with other people [1].

We can say that, emotion is the feeling state of ours as the output of our internal state, based on biochemical and environmental effects. Emotions are the focus of many disciplines, such as psychology, social sciences, sociology, philosophy and even economics. Classifying emotions into categories like joy, fear, anger, etc. is a hard topic and studied by many researchers. Paul Ekman, an important psychologist, has created a list of basic six emotions and this list is named as Ekman's List [2]. The six basic emotions listed are anger, disgust, fear, joy, sadness and surprise.

Sensing the emotions from text is gaining more interest as time passes, since textual information is not only used for describing events or facts but also a good source for expressing opinion or emotional state, which makes texts a good source for sentiment and emotion analysis. Classification of the texts as generally positive or negative is the focus of sentiment analysis, and one step further of it, recognizing the particular emotion that is expressed in text

is the task of emotion analysis.

Emotion analysis of texts serves a purpose of developing computers that are able to recognize and analyse human emotion, computers that are emotionally intelligent. Market analysis, affective computing, natural language interfaces, and e-learning environments are the example applications of this field. Previous studies on emotion analysis are mainly concentrated on English texts and we aim to close the gap of analysis of Turkish texts. To the best of our knowledge, there does not exist a previous study concentrated on exactly this subject for Turkish texts.

In this thesis, the earliest form of emotion analysis study for Turkish texts is constructed by expanding the existing methods used for English text classification. In this thesis, classification of four emotions, which are joy, sadness, fear and anger are handled. Since annotated data was not available for Turkish texts, a new dataset is needed to be built with this study. This new dataset is created by combining two different types of sources. Classification process is applied by using supervised machine learning techniques. In the overall process, extensions for the Turkish language are implemented to handle the processing requirements due to the nature of Turkish language.

1.1 Emotion in Texts

Emotion classification of texts is addressing the problem of defining and categorizing the emotions that can be expressed through texts. Categories of emotions to be used in classification are decided by evaluating different aspects. Ekman's List [2] is considered to be the cornerstone in our study. On the other hand, our first data source International Survey on Emotion Antecedents and Reactions (ISEAR) [3] involves the answers of student respondents, from a questionnaire, in which they are asked to write experiences and reactions for seven major emotions; joy, fear, anger, sadness, disgust, shame, and guilt. This dataset is translated to Turkish. Another data source is Turkish fairy tales, and in the fairy tales there exist four simple and basic emotions, which are joy, sadness, fear and anger. Since these four emotions are the intersection of all these sources, it is decided to classify texts with these emotions.

1.2 Turkish Language

Turkish is one of the morphologically rich languages (MRL) with its agglutinative structure. That means most of the words are constructed by adding suffixes to the roots of the words. Morphology of the language, decides the rules of language on creation of the word. With the same root, one can create many words that have different meanings from each other. This structure, provides to form larger number of words from a single root, and causes Natural Language Processing (NLP) task harder then other languages. It is not possible to gather all the word forms to build a lexicon to be used in NLP for MRL languages, and theoretically infinite number of words can be created. English, for instance, is a morphologically poor language, and building a lexicon is more feasible. Turkish needs a morphological parser to divide to word into its components. Such a parser, generally may not return one answer, because there exists more then one possible constructions of the words. Each result will include a root and a sequence of inflectional and derivational suffixes. This is called morphological ambiguity of the word. Some other examples of MRL languages are, Arabic, Hebrew, Czech, and Basque.

To give an example on the agglutinative structure of Turkish, suppose the word ”söyletemedim”, which is also a sentence. In Turkish it is just one word, however if we translate that to English, it is, ”I was not able to make her tell”. This example shows the importance of the suffixes, that they may give many different meanings to the word.

The standard approach in emotion and sentiment analysis of English texts is removing all suffixes of words and considering roots as the smallest parts of words. However, such a method may not be appropriate for Turkish language, because of its agglutinative structure explained. Each suffix of the word must be examined for its possibility of changing the word's meaning. Different meanings of roots may change the emotion of the statement, thus needs to be examined carefully. However, morphological ambiguity of words makes this process more complicated and causes the system to be less effective.

1.3 Contribution and Organization of Thesis

Our contributions in this thesis can be explained as follows;

- We developed a framework for analysis of Turkish texts for emotion classification. To achieve our goal, a new dataset is created and introduced for emotion classification.
- Existing approaches that were applied to English texts are extended to get better results with a morphologically rich language, Turkish.
- Different feature selection and feature weighting approaches are combined to increase the success of the system.

The rest of this thesis is organized as follows. In Chapter 2, a review on the related studies existing in literature are given. Theoretical background information on the methods used in this study is given in Chapter 3. In Chapter 4, all the phases of our study are explained in detail and experimental results are discussed in Chapter 5. The thesis report is concluded with final remarks and future work in Chapter 6.

CHAPTER 2

LITERATURE SURVEY

This study aims to improve emotionally intelligent computer programs that can detect the emotion in a sentence or in a paragraph in Turkish language. The studies conducted on this topic are varied on the type of methods used. Information on these studies is given in Section 2.1. The studies on various analysis of Turkish texts are summarized in Section 2.2.

2.1 Emotion Analysis of Texts

One of the methods widely used is the keyword spotting approach on the subject of emotion analysis. In keyword spotting, a certain list of emotion words are prepared for the emotions like happy, sad, angry, afraid, etc. The document's emotion is predicted by searching the emotional words in the document. Eliot's study [4] searches for 198 emotional words and affect intensity words. Olveres et al. [5] and Strapparava et al. [6] also used keyword spotting technique. The problem with this approach is its poor results with negated sentences and indirect emotional sentences such as "I don't know what to say, I am going to divorce today." Orthony's Affective Lexicon [7] is an example of emotional lexicons that can be used in keyword spotting.

Boucouvalas [8] developed the Text-to-Emotion Engine by using word tagging and analysis of sentences, which is a rule based approach. Ekman's six basic emotions [2] are used in the system and the system analyses the text in the chat environment and predicts the emotion communicated. In the rule-based approach, a set of rules are applied to the text and a score is calculated according to the level of emotion. Neviarouskaya et al. [9] also applied the rule-based approach for classification in on-line communication environments. Since lan-

guage is evolving every day, rule-based approaches generally fall behind. And also in the systems that are developed for on-line communication texts, problems occur because of the style of the texts, since they are in their specific forms. Masum [10] also proposed a rule based method, providing a deeper analysis. In [10], for each sentence, subject-verb-object triplets are extracted. Adjectives and adverbs are added as attributes of triplets if exist. A lexicon consisting of word-valence pairs is used in the study and by applying a set of rules to the triplets a valence for the sentence is calculated. The valence of the sentence determines the sentiment of the sentence. OCC (Ortony, Clore, Collins) model [11] is used as the emotion model, which contains emotions such that, happy for, sorry for, hope, fear, etc. The model is generated on the idea that people experience emotions in response to events, agents and objects. And, in the study the triplets are evaluated using this model. A set of rules is defined, such that if the sentence has self reaction of ”displeased” and other presumption of ”undesirable”, if the direction of emotion is ”other” and if the valence of the agent is positive, then the emotion of the sentence can be described as ”sorry for”. In another words, sorry for means being ”displeased about an event undesirable for a liked agent” [10]. A semantic parser is used to detect the agents and events. Another study [12] also used the same approach for emotion extraction.

Liu et al. [13], demonstrated an approach that uses a large scale real world knowledge about the inherent affective nature of everyday situations to understand the underlying semantics of knowledge. In the study, Open Mind Common Sense corpus [14], which is a real world corpus of 400,000 facts, is used. In the study, sentences are classified into six basic emotions in the context of an e-mail agent. Evaluation is based on user satisfaction, there does not exist a report on results.

Using statistical methods is another approach for emotion analysis. Alm [15] used children fairy tales data and classified the emotional affinity of sentences using supervised machine learning with SNoW (Sparse Network of Winnows) learning architecture. They used 30 features for their dataset, including the first sentence of the story, sentence length in words, verb count in the sentence, WordNet Affect [16] words and so on. In the study they applied 2 types of classification. The first one is, classifying as positive, negative or neutral, which is sentiment analysis. The second one is classifying as emotional or non-emotional. WordNet Affect words, which is used in the study, is another lexical list on emotions. In our study, we also generated a list of emotion words, in order to get possible emotional sentences from Turkish

fairy tales.

In another study [17], the aim is to investigate the expression of emotion in language and to prepare an annotated corpus for use in automatic emotion analysis experiments. An emotion annotation task is provided and the agreement on emotion categories ranged between 0.6 to 0.79. Cohen's kappa statistics [18] is used for measuring the agreements. In the continued study [19], a categorization of sentences into Ekman's six basic emotions [2] is done on the data collected from blogs. Corpus-based unigram features and features derived from emotion lexicons are used in machine learning. The precision values for each category ranged from 0.318 (surprise) to 0.824 (fear).

Strapparava and Mihalcea [20] applied several knowledge based and corpus based methods for the automatic identification of six basic emotions. Emotion analysis of news headlines is the basic focus and a data set of news titles is annotated to construct a dataset. A variation of Latent Semantic Analysis (LSA) is implemented and results are evaluated. Different emotion categories resulted in different success levels, some did better with LSA model, some did with naive Bayes training classifier.

To recognize emotions in news headlines, Katz et al. [21] applied a supervised approach. In the training data, each headline is scored on a scale of 1-100 with one of the predefined set of six emotions (Anger, Disgust, Fear, Joy, Sadness, and Surprise). Then, each word in headlines is lemmatized and a valence score is calculated by using the scores that are given in the training process. This study did well on the SemEval Affective Text task [22]. SemEval is evaluations of computational semantic analysis systems and Affective Text Task aims to explore the connection between lexical semantics and emotions.

Calvo and Kim [23] proposed a dimensional approach that can be used for visualizing emotions in a psychologically meaningful space and for detection of emotions. They compared their approach with the statistically driven techniques. It is stated that emotions are better represented in a 3 dimensional space of valence, arousal, and dominance. Four datasets are used in the study, SemEval Affective Text data [22], ISEAR dataset [3], fairy tales and USE (Unit of Study Evaluations). The comparison between dimensional and statistical approach does not provide a clear result, results vary among datasets.

Using machine learning systems with a training corpus of n-grams increases the success of

the system. With the training process it is possible to learn the valence of arbitrary keywords. For example if the word ”dentist” is used in many fearful texts, machine learning will be able to identify that ”dentist” has a high valence of fear.

Danisman and Alpkocak [24] used ISEAR dataset and applied Vector Space Model (VSM) for classification. Five emotion classes, which are anger, disgust, fear, sadness and joy, are used. Classification by using Naive Bayes, Support Vector Machines and Vector Space Model classifiers are compared. Stemming and stop word removal strategy is also implemented and the system is evaluated with 10 fold cross validation on ISEAR data. For the vector representation, tf-idf weighting is applied. Also to improve the success, emotional words from Word Net Affect and WPARD are added to the training dataset. With the use of all data sources, this study reached 70.2% accuracy with 5 classes. Another study [25] also used ISEAR dataset for classifying 5 emotion classes, joy, anger, fear,disgust, and guilt. The effect of lemmatization and stemming on emotion analysis is investigated. N-gram approach is implemented for features and also Weighted Log Likelihood Ratio (WLLR) scoring is applied to select the most valuable features. Naive Bayes and Multinomial Naive Bayes (MNB) are used for classification, frequency count weighting is used as feature values, and results are measured with 10-fold cross validation. They reached the accuracy of 70.4% with 400 unigrams, 400 bigrams, 100 trigrams selected as features from each class with WLLR scoring. The best accuracy is obtained with MNB classification. These two studies [24, 25] differ in classification technique, feature selection approach and feature weighting method but they reached similar accuracies. In our study, we also covered all the steps that are included in both studies [24, 25].

2.2 Studies on Analysis of Turkish Texts

The other important part of our study is working with Turkish texts. To the best of our knowledge, there does not exist a study on emotion analysis on Turkish texts. In general, the number of studies on analysis of Turkish texts is very limited. [26] has worked on sentiment analysis of Turkish texts. He collected data from a Turkish movie review website in which the users rate the movies and write comments on movies. The data is suitable for supervised sentiment analysis, since the comments are already labeled. He applied SVM to data for classifying data as positive or negative. The effects of using stemming, n-grams and part-of-speech tagging

are examined. Zemberek (Turkish NLP library) is used for stemming. For selecting the features, a threshold value is decided, and features with occurring less than the threshold value is eliminated. Unigrams, bigrams, trigrams and combinations of these are tried, and presence - non presence approach is used in feature vector preparation. On the overall, about 85% accuracy is supplied on classification of positive and negative texts.

Another thesis work, given in [27], has focused on the relation between Turkish language usage in texts and psychological states of the people who write the texts. Depressive, non-depressive, anxious and non-anxious people has written some texts and these texts are classified by using Weka tool. Zemberek is used as the morphological analyser, personal pronoun usage and the tense selected in the writings are examined according to the psychological states of the writers.

There exists some other studies on Turkish texts, such as a question answering system in Turkish using text mining techniques [28], and detection similar Turkish news with text mining [29]. These studies also include a preprocessing phase for the texts with Zemberek, and then a keyword weighting process and then cosine similarity calculation for categorizing the texts. Another study, given in [30], combines the usage of n-grams with latent semantic analysis to reach better results in document classification in Turkish. A new event detection and topic tracking system [31], is also introduced in Turkish.

CHAPTER 3

BACKGROUND

This study mainly involves Machine Learning (ML) methods for classification, Natural Language Processing (NLP) for processing the words of the texts, feature selection methods for gathering the most valuable features and feature weighting methods for reaching the most successful results. In this work, WEKA (Waikato Environment for Knowledge Analysis) software [32] is used for applying ML methods, Zemberek library [33] is used for NLP and Weighted Log Likelihood Ratio (WLLR) ranking, n-gram approaches are used in feature selection. This chapter presents information on these topics.

3.1 Classification Methods

Machine Learning, is a branch of artificial intelligence, concerned with implementation of software that can learn from past experience, and that can teach itself to change with new data. A machine learning system starts with a training data, extracts knowledge from that data, which means acquiring structural descriptions from those samples, then uses that information to predict the output of new data. Machine learning is the intersection of computer science and statistics, and is applicable in many areas like speech recognition, spam detection, computer vision, gene discovery, robotics, etc. There are various categories of machine learning algorithms such as supervised, unsupervised, semi-supervised, reinforcement learning, transduction and learning to learn. These algorithms are categorized according to the type of process desired. In this thesis, we use classification task, therefore supervised learning methods have been employed. In supervised learning, input data is a labeled training set of data and the algorithm produces an inferred function by analysing that training data. When new data is asked to be classified, the algorithm provides the answer with that inferred function

extracted in the training process.

In this thesis, in order to apply machine learning, first features of the model should be selected and then data should be represented as a vector of numerical values of that features. Applied methods for feature selection and assigning values for those features are explained in Section 3.1.4. Support Vector Machines, Bayesian algorithms are the two types of supervised algorithms that are applied to training and testing sets of data in this study.

3.1.1 Naive Bayesian Classifier

Naive Bayesian (NB) Classifier [34] is one of the most popular classifiers that are used for text classification. Since it is efficient, easy, and is applicable to high dimensional data, it is widely used. The basic idea is, based on Bayes' theorem, using the joint probabilities of classes in the dataset to predict the new classes. Assume we have an element $x = (f_1, f_2, ..f_m)$, and f_i is the value of one of the m features. Then the probability of x to be in class c_i, can be represented as;

$$p(c_i|(f_1, f_2, ..f_m)) = \frac{p((f_1, f_2, ..f_m)|c_i)p(c_i)}{p(f_1, f_2, ..f_m)} \tag{3.1}$$

Here $p(c_i)$ is the probability of class c_i, which is calculated by the number of samples in class c_i divided by the number of all samples count. Since $p(f_1, f_2, ..f_m)$ does not depend on the class, the denominator can be discarded. Then, the formula can be rewritten by using chain rule, with having the assumption that the presence of a feature of a class is independent from the presence of any other feature, given the class, as;

$$p(c_i|(f_1, f_2, ..f_m)) = p(c_i)p(f_1|c_i)p(f_2|c_i)p(f_3|c_i)...p(f_m|c_i) \tag{3.2}$$

So, the posterior probability of the testing sample for each class is calculated with Equation (3.2) and the class with the highest value is chosen to be predicted as in Equation (3.3).

$$class = argmax_{class} P(class) \times \prod_i P(feature_i \mid class) \tag{3.3}$$

Independence assumption of features is the naive part of the algorithm. Due to this limitation, algorithm may result in poor assumptions.

3.1.2 Complement Naive Bayes

Naive Bayes (NB) algorithm, explained above, is a basic technique in text classification; however the assumptions of the algorithm decrease the quality of the overall classification process. The problems of Naive Bayes are, assumption of feature in-dependency and unbalanced training data bias [35]. Uneven size of training samples, that is one class having more training samples than the others, may cause NB to erroneously prefer one class over the other, because it causes the decision boundary weights to be biased. Complement naive bayes (CNB) is a modified version of naive bayes algorithm to improve the accuracy under unbalanced training datasets. CNB algorithm uses data from all classes except the class which is focused on (denoted with $class'$), while learning the weights. Equation of Naive Bayes, (3.3), can be written as in (3.4) if we take the log-likelihood;

$$class = argmax_{class}(\log P(class) + \sum_i P(feature_i \mid class)) \tag{3.4}$$

Then considering complement classes, c', rather then the class itself, the formulation is converted to;

$$class = argmax_{class}(\log P(class) - \sum_i P(feature_i \mid class')) \tag{3.5}$$

The equation can also be expressed as;

$$class = argmax_{class} P(class) \times \prod_i \frac{1}{P(feature_i \mid class')} \tag{3.6}$$

3.1.3 Support Vector Machines

Support Vector Machines (SVM) are introduced by V.Vapnik [36] and commonly used in text classification. A set of labeled training data is given and the algorithm constructs a model

using that data. The model is the mapping of input vectors into high dimensional feature space. A linear decision hyperplane is then constructed to find the optimal separation between the classes. This optimal separation means the biggest clear gap between the classes. To find the optimal surface Structural Risk Minimization (SRM) [37] principle is used. SRM is used for minimizing the error in separation of the training data. It is said that to find the optimal hyperplanes, only a small amount of training data is needed, which are called the support vectors. Support vectors are the vectors that constrain the optimal hyperplane. In Figure 3.1 support vectors are marked with grey squares. The testing samples are then mapped into the same space and the classification is done by looking at the place of the item according to the decision surface.

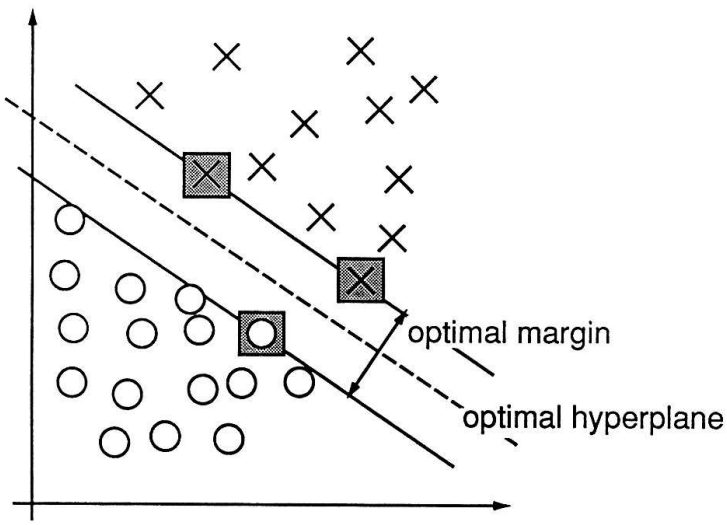

Figure 3.1: An Example of Separation in 2d (Figure taken from [36]).

Let $w \cdot x + b$ be the equation of the optimal hyperplane. w is a vector that is perpendicular to the plane, x is a input vector in the feature space an b is a constant. Then the linear decision function, $I(x)$ is defined as $sign(w \cdot x + b)$. w can be written as a linear combination of training vectors;

$$\sum_{i=1}^{l} \alpha_i y_i x_i \tag{3.7}$$

where l is the support vector number, x_i is the support vector, $y_i \in 1, -1$ is the class indicator and α_i is the Lagrange multiplier. Then the decision function can be rewritten as;

$$I(x) = \sum_{i=1}^{l} \alpha_i y_i x_i \cdot x + b \tag{3.8}$$

SVM is a non-probabilistic binary linear classifier. For classification in multi-class, multiple binary classifications on data are applied. Two strategies can be used; one-versus-all, which is classification between one of the labels and the rest, or one-versus-one, which is classification between every pairs of labels. In this thesis LibSVM library [38] and SVM-multi-class, an improved version of SVM-light [39], are tried, and all gave almost the same results. So, it is decided to run the tests with LibSVM. In LibSVM, one-versus-one approach is implemented. That is, for each pair, a binary classification is done and the winner gets a vote. After all pairs are classified, the class with the highest vote is the actual winner. If there exists k number of classes, then $k(k-1)/2$ classifiers are created and each classifier has 2 classes of data to train.

3.1.4 Feature Selection and Feature Vector Construction

In text classification process, data is represented as a set of feature vectors. The straightforward approach of feature set is to be the distinct set-of-words in the dataset, and treating the text as a bag of words (BOW). The next approach is using word sequences as features which is called n-grams. Also, rather than using all set of words or all set of word sequences in text classification process, selecting the features that best describes the dataset is an important part. To be able to separate the classes from each other successfully, informative features should be extracted from data. For this purpose, weighted log likelihood ratio (WLLR) scoring is used. In this section, n-gram approach and WLLR is explained.

3.1.4.1 N-Grams

N-Grams are the sequence of words of length n. In bag of words approach, the text is treated as a collection of words, and the order of words are not regarded. For example, ”Sue is nicer than Mary” and ”Mary is nicer than Sue” is the same in BOW approach. This causes a big information loss on the text. Using n-gram model reduces the information loss with the use of ordered set of words as features [40]. An example set of n-grams can be seen in Table 3.1. Unigrams are words, bigrams are the two word phrases, trigrams are three word phrases and so on. A number of studies are conducted on the outcome of using n-grams on text

Table 3.1: Example Set of N-grams

Sentence	She took my stuff without permission.
Unigrams	{She},{took},{my},{stuff},{without},{permission}
Bigrams	{She took},{took my},{my stuff},{stuff without},{without permission}
Trigrams	{She took my},{took my stuff},{my stuff without},{stuff without permission}
Fourgrams	{She took my stuff},{took my stuff without},{my stuff without permission}
Five-grams	{She took my stuff without},{took my stuff without permission}

classification, and [41] has resulted that using ngrams decrease the performance. However, later researches has showed the opposite; [40] and [42] concluded that, using bigrams and trigrams as features rather then using just words as in BOW approach, improve classification performance, whereas longer n-grams do not bring any success.

3.1.4.2 Weighted Log Likelihood Ratio

On the topic of selecting best features, Nigam [43] has showed that ranking features with WLLR gives good results. Each feature's WLL ratio is calculated as in Equation (3.9), and the features with the highest scores are the ones that are the most distinctive features over all classes.

$$WLLR(w_t, c_j) = P(w_t \mid c_j) \times \log(\frac{P(w_t \mid c_j)}{P(w_t \mid \neg c_j)}) \tag{3.9}$$

where;

- w_t is the t_{th} feature whose scores is being calculated,
- c_j is the j_{th} emotion class,
- $P(w_t \mid c_j)$ is number of appearances of w_t in c_j divided by number of all features in c_j,

- $P(w_t \mid \neg c_j)$ is number of appearances of w_t in $\neg c_j$ divided by number of all features in $\neg c_j$.

By using Equation (3.9), a feature in class j will have a high rank, if it appears frequently in class c and infrequently in classes other than c [44]. So, the features that are the most distinctive ones can be extracted and these features would best represent data in the vector space.

3.1.4.3 Feature Vector Construction

As stated earlier, data is needed to be represented as feature vectors; this means for each document a row of numerical feature values should be generated. Three different approaches are applied in this study while assigning values to features. Presence-nonpresence, term frequency (tf) and tf-idf (term frequency-inverse document frequency) are the methods. Presence-non presence is used in BOW, and is simply giving 1 or 0 according to the presence of the feature in the document [45]. Frequency count (term frequency) is another simple method, where the value is the number of the occurrences of the feature in the document. The frequency count of term t in document d is denoted as $tf_{t,d}$. The last method, term frequency-inverse document frequency (tf-idf), is a slightly more complex, and it is calculated as in Equation 3.10a. Here, inverse document frequencies are also taken into account to give high weights to the terms that occur many times in small number of documents [46]. Document frequency, df_t, is defined to be the number of documents that contain the term t. Inverse document frequency of an item is the logarithm of ratio of the number of all documents to the number of documents that contain term t, and is calculated as in Equation (3.10b), where N is the total number of documents. By looking at the formula, we can say that idf of a rare term will be high, however idf of a common term will be low. Tf-idf of a term is the multiplication of tf and idf of the term. This formula suppose to increase the discrimination of the documents.

$$tf - idf_{t,d} = tf_{t,d} \times idf_t \tag{3.10a}$$

$$idf_t = \log(\frac{N}{df_t}) \tag{3.10b}$$

3.1.5 WEKA

Weka is the project that collects together the machine learning methods [32]. It is an open source software, written in Java language and provides several data mining tasks with a user friendly graphical user interface. In this thesis, we used Bayes and SVM functions in Weka. The data to be classified is created in one of the formats of CSV, LibSVM's format, C4.5 or WEKA's own Attribute Relationship File Format (ARFF) to be stored in a database. In the arff format, there are two sections, the first one is the header section in which the relation name, attributes and attribute value types are stated, the second one is the data section, where each line stores the attribute values of the entry. An example view of Weka can be seen in Figure 3.2.

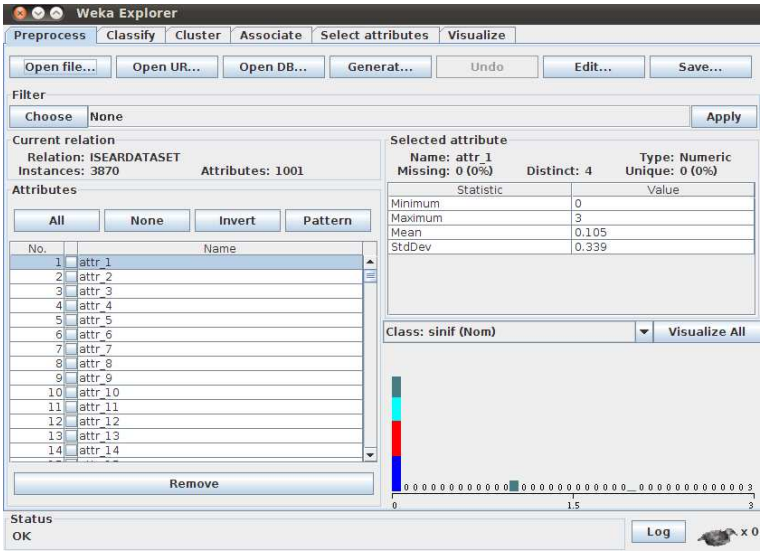

Figure 3.2: A Screenshot of Weka

Weka provides cross validation on the selected learning algorithm which is a very useful facility for evaluation. In the n-fold cross validation technique, data is randomly divided into n subsets, n-1 subsets are used as training data, the remaining subset is used for testing and this process is run for n times. The output of n processes is averaged to get the most realistic result. In Weka's 10-fold-cross-validation policy, at first all data is randomized. Then data is stratified, in order to get almost the same class distribution for all the training sets as in the full dataset. After that, training and testing sets are generated. While randomizing data, the random value depends on the seed value that is given manually. By this way, repeatable

results from the same dataset are allowed.

3.2 Natural Language Processing

Natural Language Processing (NLP) is the research area of computer science for understanding and analysing human natural language. It can be said that NLP task is human-like language processing, and enclosed in artificial intelligence [47]. An NLP system may be able to translate a text to another language, or answer questions by processing the content of the question, or paraphrase the given text. Of course, there are many other goals that a system would try to reach. The main steps of NLP are morphological analysis, syntactic analysis, semantic analysis, and discourse analysis. Morphological analysis is analysing each word with its meaning, suffixes and prefixes, that is analysing structure of the word. Syntactic analysis is analysing the grammatical structure and relations of the words with each other. In semantic analysis, structures created by syntactic analysis are assigned meaning. Discourse analysis deals with the preceding sentences that affect the meaning of next sentence.

The area of NLP is board and is being used widely over the decade to make human life easier. With internet usage, everyone can express his/her opinion on anything to other people. This information is a valuable resource for many purposes. Sentiment analysis and opinion mining studies deal with those information to examine the document whether it is subjective or neutral and if it is subjective, examine the document whether it is positive or negative. These studies are one of the most famous subfields of NLP. One step further of sentiment analysis is emotion analysis of text and automatically capturing emotions of text is gaining more attention on NLP applications.

Combining NLP and ML techniques for emotion classification of texts is proven to give good results. For the emotion classification purpose, each word of the document is analysed morphologically with an NLP tool to extract some information about the word. Also, part-of-speech (POS) tagging is provided by NLP tools. Part-of-speech tagging is marking each word in the sentence with one of the corresponding categories such as nouns, verbs, adjectives or adverbs. In English, there exists so many NLP tools that uses the sentence structure to perform POS tagging. Whereas in Turkish there is only one popular NLP tool, Zemberek [33], that can do morphological analysis and POS at word level.

```
Kelime = "olamaz"

çözümlemeler:
 [ Kok: ol, FIIL ]  Ekler: FIIL_YETERSIZLIK_E + FIIL_OLUMSUZLUK_ME +
FIIL_GENISZAMAN_IR
 [ Kok: ol, FIIL ]  Ekler: FIIL_ISTEK_E + FIIL_OLUMSUZLUK_ME + FIIL_GENISZAMAN_IR
 [ Kok: o, ZAMIR ]  Ekler: ISIM_DONUSUM_LE + FIIL_OLUMSUZLUK_ME +
FIIL_GENISZAMAN_IR
 [ Kok: o, ZAMIR ]  Ekler: ISIM_DONUSUM_LE + FIIL_DONUSUM_MEZ
```

Figure 3.3: Example Analysis of Zemberek

```
Kelime = arkdaş
öneri:
      arkadaş
      arklaş
```

Figure 3.4: Suggestion of Zemberek

3.2.1 Zemberek

Zemberek is the most famous NLP library for Turkish. It is an open source Java library. Zemberek provides morphological analysis and spell checking functions for Turkish words. With the use of Zemberek library, words are examined morphologically, negations are handled, some important suffixes are extracted because of their important contribution in meaning. Since Turkish is an agglutinative language, with the use of affixes a word may have a totally different meaning, and so, examining affixes is a crucial task not to have information loss. Zemberek gives all possible results for the morphological analysis of a word. Due to the agglutinative structure, more than one result for a word (ambiguity) may appear. The result set contains all possible root-affix combinations in a decreasing order of possibility. So, the highest possible root-affix combination of a word is the first item in the solution set of Zemberek. An example analysis with the word ”olamaz” is shown in Figure 3.3. Three possible results are provided for the word and the first one is the correct one.

For the words that may be mistyped, Zemberek provides a function for checking the word and also, if it is not a correct word, a function for suggesting a word instead. An example can be

seen in Figure 3.4.

CHAPTER 4

PROPOSED METHODS

Our system mainly involves five phases; data gathering, preprocessing data, feature selection, feature vector construction and classification. A general information on the phases is as follows:

- Data gathering is the process of collecting suitable data for our analysis. This phase involves translating ISEAR dataset to Turkish, collecting data from Turkish fairy tales, annotating fairy tales data and combining two datasets.

- Preprocessing includes removing unnecessary parts of text and providing the most meaningful smallest part of words by examining negations and special suffixes.

- Feature selection is the next phase, different approaches are tried, such as selecting all distinct words (unigrams) and selecting n highest scored features with Weighted Log Likelihood Ratio (WLLR) scoring. Different combinations of n-grams are tested.

- Features are weighted by using both tf and tf-idf methods.

- Classification is the whole process of training and testing parts. Different classifiers are tested, with analysing best parameters.

The general overview can also be seen in Figure 4.1. In each phase different approaches are tried to get the best results. Java programming language is used for implementation and eclipse is chosen for the development environment. Data is stored in a MySQL database. Each phase is described in more detail in the following sections.

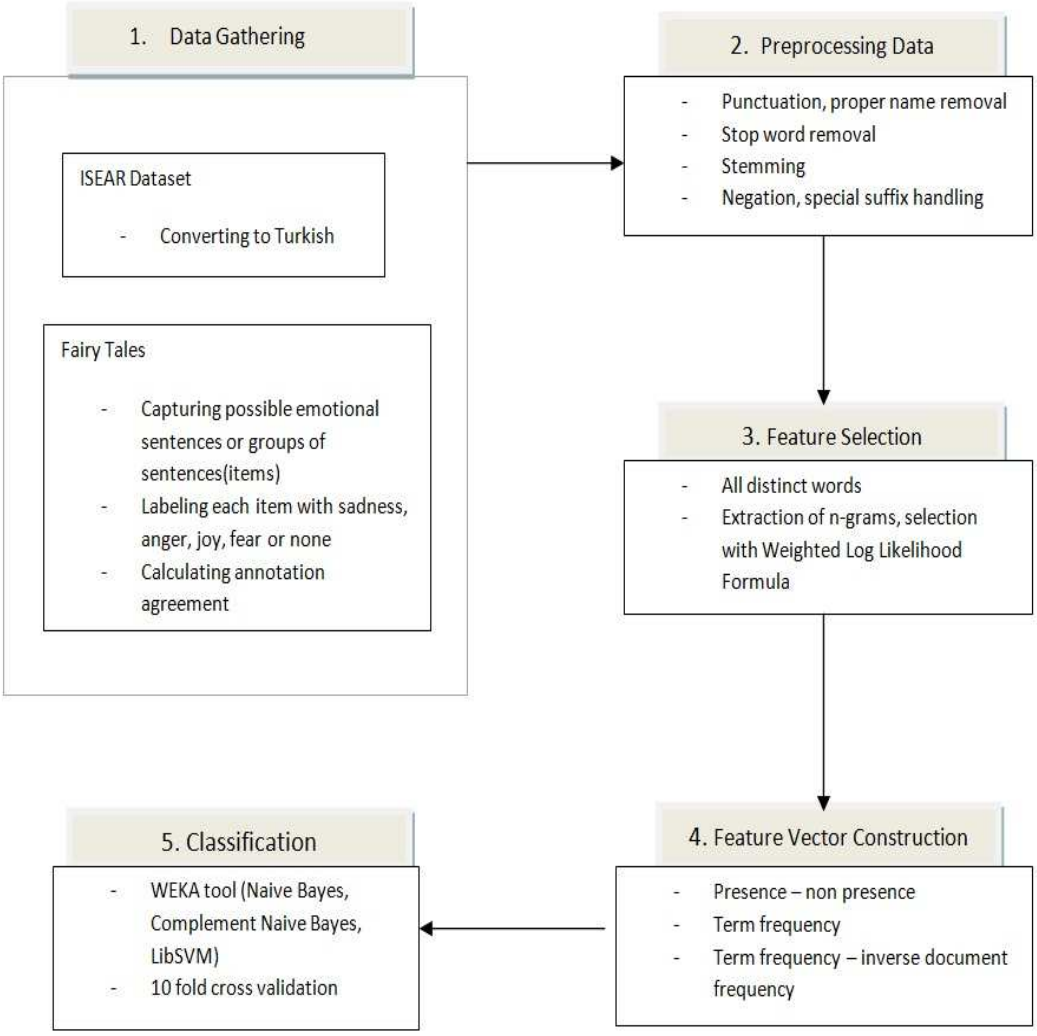

Figure 4.1: Overview of the System

4.1 Data Gathering

Two types of data sources are used in this study. The first one is International Survey on Emotion Antecedents and Reactions (ISEAR) [3] dataset and the second one is data collected from Turkish fairy tales [1]. Properties of datasets and the methods used to construct them are explained in this section.

[1] http://www.bilgicik.com/yazi/masallar/, http://www.masaldinle.com/, last accessed 15.07.2012.

4.1.1 ISEAR Dataset

ISEAR is mainly a project, directed by Klaus R.Scherer and Herald Walbott. 3000 people from 37 countries are involved in the study and these people are asked to write the situations that they experienced 7 major emotions and reactions to those emotions, which makes this dataset is suitable for emotion classification purpose. There exists some studies that use this dataset for this purpose [24, 23, 25]. It is decided to translate this dataset to Turkish, to initiate the study of emotion classification. This translation is a hard task, because the styles of the writings are different from an ordinary Turkish writing and also the form of a Turkish person emphasizing his/her feeling is so different than the existing forms in the dataset. We collected a team of 33 people for the translation task, and give information to them on how to do the translation and on what points they should be more careful. They are told that, they are not expected to translate the sentences to Turkish directly, but they are expected to read and understand the answer, then try to give the answer in their style of Turkish. By this way, we aimed to construct the most realistic Turkish dataset. After the participators returned their results, all the results are controlled for their reliability. Meanwhile, it should be noted that the typing mistakes in the data are not corrected manually. Zemberek library [33] is used to correct these words and this process is explained in Section 4.2. There exist some sentences in the dataset that are included in more than one classes, since some people give the same answers for different emotions. We did not eliminate those sentences to have a dataset that is similar to the original version. Some samples of the original dataset and translated versions can be seen in Table 4.1.

Original dataset contains 7 emotions in total, and 4 of them are translated to Turkish for this study. Data distribution of the resulting set can be seen in Table 4.2. Our version of dataset consists of 4265 items in total, 1073 from joy, 1036 form sad, 1083 from anger and 1073 from fear emotions.

4.1.2 Turkish Fairy Tales Dataset

The other data source of ours, fairy tales, are collected from several web sites. Namely, we have used 25 children fairy tales in our study and the list of these can be seen in Appendix A.2. Since we are building a sentence and paragraph level emotion analysis, we have to divide

Table 4.1: Samples from ISEAR Data

Having passed an exam Bir sınavdan geçtiğimde	joy
When I met a good friend of mine after a long time. Uzun bir aradan sonra yakın bir arkadaşıma rastladığımda	joy
Saw poverty in the countryside. Kırsal bölgelerde gördüğm yoksulluk.	sad
A young, close relative of mine died, leaving behind a baby a few months old. Genç ve yakın bir akrabam bir kaç aylık bebeğini arkasında bırakarak vefat ettiğinde.	sad
One day I lent my tennis I just had washed to my sister because she asked it. I asked her not to soil it as I had just washed it. Next day I looked at the tennis, and it was dirty with wax. She could not have soiled it. It was lack of consideration. I felt very angry. Ayakkabılarımı kardeşim çok istediği için ödünç verdim. Ona iyi bakmasını yeni yıkadığımı söyledim ama geri getirdiğinde ayakkabılar çamur içindeydi. Çok ama çok sinirlendim	anger
When my sister took something that belonged to me without my permission. Kız kardeşim benim eşyalarımı izinsiz kullandığında.	anger
While preparing my master's thesis, I was scared that I would not accomplish anything as the subject was rather difficult. Yüksek lisans tezimi hazırlarken. Konu çok zor olduğundan hiçbir şeyi başaramayacağımdan korktum.	fear
While paddling in the river during a storm. I feared drowning. Fırtınalıbir havada nehirden geçerken.Boğulmaktan korkmuştum.	fear

Table 4.2: Distribution of ISEAR Dataset

Class	Count	Dataset
joy	1073	
sad	1036	
anger	1083	ISEAR
fear	1073	
total	4265	

Table 4.3: Examples from Emotional Words List

Emotion	Seed Words
Fear	dehşet, endişe, kaygı...
Joy	mutlu, keyif, kahkaha...
Sad	çaresiz, ağla(mak), dert...
Anger	öfke, asabi, zorba...

the text into segments, but these segments are not considered simply as sentences. If the text includes quotations, the segment should be the whole item that contain the quotation. Using this approach, texts are divided into items. Throughout the thesis, the word ”sentence” is used, but it covers the meaning of ”item”.

To construct our dataset, the sentences should be annotated manually. For this process, the first thing is to get the sentence level texts, which are possibly emotional, to be annotated. To get the possibly emotional sentences, a new list of emotional Turkish seed words is constructed, and the sentences that contain a word from this list are thought to be emotional. Some sample words from our seed word list can be seen in Table 4.3.

The sentences that do not contain a word from the whole list are filtered out and the resulting set of sentences are given to our annotators.

4.1.2.1 Data Annotation Process

After collecting the sentences, the annotation process starts. Three people worked on the annotation process and the sentences are labeled with one of the 5 classes, which are joy, sad, anger, fear and none. The class named as ”none” is important since the sentence may

Table 4.4: Samples from Fairy Tales Data

Anne kız uzun yıllar mutlu bir şekilde, beyaz evlerinde, güzel çiçekleri ile yaşamaya devam etmişler.	joy - high
’Belki de bu yaşama alışırım,’ diye düşünmüş, neşesi yerine gelmiş azıcık.	joy - low
Eve geldiklerinde babalarının biricik kızını karşıları nda görünce kıskançlıktan ve öfkeden çatır çatır çatlamışlar.	anger - high
Fakat derste yaptıkları davranıştan dolayı öğretmenin kızacağını düşündüler.	anger - low
Bu çok sevgili arkadaşına yalancı çıkmak, haramilerin yapacağı kötülükten daha fazla üzmüş kendisini.	sad - high
Babasının acıkacağını, yiyecek bir şey bulamayacağını, gecikirse anneciğinin merak edeceğini düşünüyormuş.	sad - low
Öylesine korkunçmuş ki, tüccar neredeyse korkusundan bayılacakmış.	fear - high
Kadın ürkek adımlarla odadan odaya dolaşıyordu.	fear - low
Nilüfer perisinin bir an önce gül perisini bulması gerekiyordu.	none

not have any feeling, even if it contains a word from our emotional seed words list. Also, the levels of emotional states of the sentences, as high, medium or low, are labeled by these people. Annotation task is hard because of the subjectivity of the process. On the same sentence, two people may have different opinions, especially on the decision of level, that is one annotator thinks that the sentence should be classified as joy high, while the other one thinks that the sentence should be classified as joy low. Three people labeled the same sentences, and Cohen's kappa statistics is used as the measure of annotation agreement.

Labeling Agreement and Cohen's Kappa

After the labeled sets are collected from different people, the measure of agreement is calculated. Cohen's Kappa [18] is the most popular measure of agreement used to compare the decisions of two judges. It calculates a value by considering not only proportional agreement, but also the amount of agreement that would be expected just by coincidence. To give an example, the following table is the annotation results of two annotators, J1 and J2, for our two classes, joy and none.

		J2	
		joy	none
J1	joy	404	46
	none	35	104

As can be seen from the table, J1 and J2 agreed on 404 items to be in joy class and 104 items to be in none class. J1 decided 46 items to be in joy class but J2 decided that those items should be in none class and so on. The value of Kappa is defined as:

$$k = \frac{p_0 - p_e}{1 - p_e} \tag{4.1}$$

Where p_0 is the observed proportion of agreement and p_e is the proportion of agreement expected by chance. If we calculate kappa from the table, the observed proportion of agreement p_0 will be $(404 + 104)/589$. J1 decided 450 items to be joy, and 139 items to be none, so decided 76% of joy; 24% of none; J2 decided 439 items to be joy, and 150 items to be none, so 75% items for joy; 25% for none. Thus, the probability of both decides joy by chance is $0.76 \times 0.75 = 0.57$ and the probability of both of them deciding none by chance is $0.24 \times 0.25 = 0.06$. The proportion of agreement expected by chance is then, $0.57 + 0.06 = 0.63 = p_e$. At last, kappa value is $(0.86 - 0.63)(1 \times 0.63) = 0.62$.

There exists a standard for evaluating the kappa value: if it is less then 0.20 it means poor agreement, if it is between 0.20 and 0.40 fair agreement, between 0.40 and 0.60 moderate agreement, between 0.60 and 0.80 good agreement and at last if it is 0.8 to 1.0 it is very good agreement. In our example, agreement level is good.

Table 4.5: Agreement Matrix of Two Annotators, J1 and J2

Class	joy	sad	anger	fear	none
joy	405	12	0	0	46
sad	4	348	13	7	21
anger	0	5	139	11	13
fear	0	12	9	120	3
none	35	24	6	2	104

Table 4.6: Pairwise Agreement of Emotion Categories

Class	J1↔ J2	J1 ↔ J3	J2 ↔ J3	**Average**	**Evaluation**
joy	0.8068	0.7697	0.8605	0.8123	very good
sad	0.7803	0.8279	0.7803	0.7961	good
anger	0.7092	0.7624	0.8171	0.7629	good
fear	0.7317	0.758	0.7778	0.7558	good
none	0.4094	0.363	0.3269	0.3664	fair

The agreement matrix of our two annotators for the emotion classes is given in Table 4.5. By looking at the table we can say that most of the confusion occurs between the class none and the others. For example, our second annotator labeled 46 of the items as ”none”, which are labeled by our first annotator as ”joy”. Whereas, there is not any item that one thinks as ”joy” and the other thinks as ”anger”. The statistical results of on the agreements of all our annotators are showed in Table 4.6 and Table 4.7. In the tables, at first, class wise measures of agreements for the emotional categories, and then measures for the levels of emotions are given. It can be seen that the agreements for categories are good, except the class none. This is due to the fact that, the highest conflict of annotators is the decision for the sentence to be emotional or non-emotional. On the other hand, the agreement on the levels of the emotions are generally low, which is an expected situation, since it is not only a highly subjective decision, but also it depends on the mood of the people. Deciding on the levels of emotions

Table 4.7: Pairwise Agreement of Emotion Levels

Class	level	J1↔ J2	J1 ↔ J3	J2 ↔ *J*3	**Average**	**Evaluation**
joy	high	0.5412	0.4995	0.5156	0.5187	moderate
	medium	0.3434	0.3778	0.26111	0.3274	fair
	low	0.3193	0.2495	0.1895	0.2527	fair
sad	high	0.4145	0.4093	0.4009	0.4083	moderate
	medium	0.2581	0.2792	0.2227	0.2533	fair
	low	0.3895	0.3393	0.3524	0.3604	fair
anger	high	0.4826	0.2589	0.2054	0.3156	fair
	medium	0.2683	0.1981	0.1282	0.1982	poor
	low	0.5	0.25	0.1071	0.2857	fair
fear	high	0.4588	0.4235	0.5694	0.4839	moderate
	medium	0.377	0.3051	0.209	0.2970	fair
	low	0.4815	0.2821	0.2857	0.3497	fair

is harder then deciding the emotion itself. This can also be seen from the Table 4.7. The agreement values are far more less then the others. In general, agreement can be considered to be fair on the levels of emotional statements.

In the overall process, the labels that are agreed by at least two people are considered as true and these sentences are involved in the dataset. Some samples from fairy tales dataset can be seen in Table 4.4 and the total number of each classes are given in Table 4.8. The counts of high, medium, low levels of Turkish fairy tales data are also shown in Table 4.9.

Table 4.8: Distribution of Fairy Tales Data

Class	Count	Dataset
joy	408	Fairy Tales
sad	368	
anger	149	
fear	124	
none	112	
total	1162	

The two data sources, ISEAR and fairy tales, are then combined to have a more complicated

Table 4.9: Fairy Tales Emotion Levels

Class	# high	# midium	# low
joy	187	129	79
sad	182	112	62
anger	77	50	15
fear	64	33	20

data set. The class ”none” in the fairy tales dataset is discarded in the combined set and the resulting distribution is as in Table 4.10.

Table 4.10: Distribution of Combined Data

Class	Count	Dataset
joy	1481	
sad	1404	
anger	1232	Combined Dataset
fear	1197	
total	5314	

After all, ISEAR dataset contains 4265 items, 1073 from joy, 1036 from sad, 1073 from fear and 1083 from anger. Fairy dataset contains 1161 items in total, 124 of fear, 149 of anger, 408 of joy, 368 of sad and 112 of none. In ISEAR dataset, each class has almost the same amount of data, whereas fairy tales dataset classes are not uniformly distributed. The combined dataset, contains 5314 items in total.

4.2 Data Preprocessing

In the preprocessing phase, punctuations and proper names are removed, morphological analysis and spell checking of the words is applied and stop words list removal is done. Some exceptions are needed to be implemented for some words. Zemberek library [33] is used for morphological analysis.

Punctuations and proper names are not needed and removed from data since they do not provide useful information in our analysis. The emotion in the sentence is not related to the proper names and removing them increases the efficiency of the system. The proper names are

detected with a simple approach. If the word is in the middle of the sentence and starting with a capital letter, or if the word contains the character apostrophe('), then the word is thought to be a proper name and removed from sentence.

Since a word may be mistyped, spell checking of the words is needed. By using Zemberek's functions each word is controlled and if the word does not exist in the lexicon directly, then suggestion function of zemberek is used and the most possible suggestion is considered as the correct version of the word. If there does not exist a suggestion, the word is removed from the sentence. An important reason for needing a spell check is typing by using English keyboard. Authors do not type the letters "ı,ö,ü,ç,ş,ğ", and these words must be corrected not to cause an information loss. For example, if we do not correct the word "uzgun" as "üzgün" in a sentence, then the feature "üzgün" will not seem to appear in the sentence and an important emotional word will be disregarded erroneously. For this reason, checking functions of Zemberek is performing a significant job. The convenient results of a suggestion function are listed in order of their possibilities. The highest possible result is chosen to be the correct one and used in the study, whereas it is not guaranteed to be the correct one. An example view is given in Table 4.11. The first sentence is analysed correctly, but the other sentence is not.

Table 4.11: Spell Checking with Zemberek

Sentence	Annesi ona cok kizmis.
Suggestions for "cok"	çok, sok, çök, ok, kok, şok, etc.
Suggestions for "kizmis"	kızmış
After correction	Annesi ona çok kızmış.

Sentence	Aglamaktan gozleri kipkirmizi olmus.
Suggestions for "aglamaktan"	allamaktan, atlamaktan, anlamaktan, etc.
Suggestions for "gozleri"	gözleri
Suggestions for "kipkirmizi"	kıpkırmızı
Suggestions for "olmus"	olmuş
After correction	Allamaktan gözleri kıpkırmızı olmuş.

After correction process, stop word removal is applied and this process is explained in the next section.

Table 4.12: Examples from Stop Words List

pronouns	biz, siz, onlar...
numbers	on, yirmi, otuz...
conjunctions	ve, veya, ya da...
other	şey, falan, öyle...

4.2.1 Stop Word Removal

Stop words are the words that are filtered out of our data, that is they do not provide any information. Furthermore, removing these words increases the success of the system in certain cases. Such a case can be exemplified with the sentence ”mutlu falan değilim.”. Here the word ”değil” negates the word before it, which is ”falan”. Whereas, the word ”mutlu” is the one that should be negated. After removing the stop word ”falan” from the sentence, the sentence can be processed correctly.

A stop word list, which is found on the Web[2], is rearranged to be suitable for emotion classification purpose. Some words are left out, that may be important in our case, such as ”kimse, çok, niye, rağmen”. This modification in the original list increases the emotion prediction accuracy. Some example words from the list can be seen in Table 4.12. Especially, personal pronouns, numbers, conjunctions are included in the list. The whole list of our stop words can be found in Appendix A.

4.2.2 Morphological Analysis

Morphological analysis has high importance because of agglutinative structure of Turkish. The inflectional suffixes of the words should be cleaned, whereas the formation suffixes should not, because formation suffixes are the suffixes that change the word’s meaning completely. An example of the morphological analysis of a word is given in Chapter 3 Figure 3.3. As stated earlier, Zemberek gives the results in possibility order, and in this study, the first set is considered as the correct result.

[2] http://www.devdaily.com/java/jwarehouse/lucene/contrib/analyzers/common/, last accessed 03.06.2012.

Table 4.13: Samples for Negation Suffixes of Zemberek

Suffix Name in Zemberek	Suffix in Turkish	Example
FIIL_OLUMSUZLUK_ME	-me, -ma	gel**me**di
FIIL_OLUMSUZLUK_SIZIN	-sizin, -sızın	gitmek**sizin**
ISIM_YOKLUK_SIZ	-siz, -sız	huzur**suz**

Since our system is based on the word relations in data, the words that have the same root and meaning are important. The aim of morphological analysis is extracting the words with the same meaning and root, even if they appear to be different because of the suffixes they have. Stemming, which is considering the roots of each word, is the general convention in this process. We also used the roots of the words, however we added some keywords related to the changed meaning of the word.

4.2.2.1 Negation Handling

In Turkish language, negation can be given in several ways. The first one is the general way of negating a verb with a suffix. Two types of suffixes exist for negating a word. The other one is negating a noun with a suffix, which reverts the meaning of the noun to the opposite side, non existence. This is like the ”less” suffix in English, such as ”homeless”. In Turkish it is ”siz” suffix in the word ”evsiz”. Another negation type is with the word ”değil”, that is negates the meaning of the verb or the noun that it comes after. For example, the sentence ”I am not sick” can be expressed in Turkish as ”Hasta değilim”. Here ”hasta” means ”sick”, and should be negated. The last negation type is done with the word ”olmamak”. The English example ”not being sick” can be translated to Turkish as ”hasta olmamak”. In here also the word ”hasta” should be negated. All the negation types with examples can be seen in Table 4.13.

Let the word after preprocessing to be represented as ”rootOfTheWord”, then if a negation is detected with one of the suffixes of that word, the word is represented as ”_rootOfTheWord”. Also, if a negation word is detected in the sentence, then the word preceding the negation

Table 4.14: Specially Treated Suffixes

Suffix Name in Zemberek	**Example In Turkish**	**Example In English**
FIIL_YETERSIZLIK_E	gid**e**medim	I was not able to go
ISIM_KUCULTME_CEGIZ	kadın**cağız**	poor woman
FIIL_EMIR_O_SIN or FIIL_EMIR_SIZ_IN	**gidin** buradan	go away

word is changed as "_rootOfTheWord".

4.2.2.2 Handling Special Suffixes

Some suffixes in Turkish, assign important meaning to the word. The examples can be seen in Table 4.14. FIIL_YETERSIZLIK_E suffix in word "gidebilmek", means "being able to go" in English. The phrase "able to" is important in emotion analysis, since it gives a sad feeling when used with negation, like "I was not able to go". In Turkish if we do not care the suffix, the sentence would mean "I did not go", and this brings information loss. So, examining these suffixes increases the success of the system. Another suffix, ISIM_KUCULTME_CEGIZ is also giving sad feeling strongly, and at last FIIL_EMIR_O_SIN is expressing giving order to someone and is especially is used in anger emotion. When one of these suffixes occurs in the word, a special keyword is added to the sentence, regarding the suffix. The keywords added are "YT" for FIIL_YETERSIZLIK_E, "KUC" for ISIM_KUCULTME_CEGIZ and "EMIR" for FIIL_EMIR_O_SIN.

Part of speech tagging is applied for some homonym words, such as "kız". The word "kız" when used as a verb, means "to be angry", whereas when used as a noun, means "girl". To differentiate these, the noun version of the word is represented as "kızN".

Some exceptions are implemented, due to the confusion of negation suffix in some cases. For example, the highest possible correct result of Zemberek, of the word "gül**meye**" in the sentence "gülmeye başladım = I started laughing", says that there is a negation in the word. This is due to the structure of Turkish, the library analyses the word with a different approach,

Table 4.15: Example of Sentence Preprocessing

Sentence	Kızcağız bu acıya dayanamadı, kutulardan birini açıp bir kibrit çıkardı.
After Stemming	kızN bu acı dayan kutu biri aç bir kibrit çıkar
After Stop Word Removal	kızN acı dayan kutu aç kibrit çıkar
After Negation Handling	kızN acı _dayan kutu aç kibrit çıkar
After Special Suffix Handling	KUC kızN acı YT _dayan kutu aç kibrit çıkar

as in the sentence ”gül**meye**sin = you should not laugh”. However, the use of second approach is not very common in the language, therefore, we defined an exception so that for the words ending with ”meye” and ”maya”, negation is not handled even if the best result of Zemberek says so. Another exception is applied for the words ”korkut-(mak)” and ”korku”. When stemming is applied on the word ”korkut-tu”, stem is given to be ”korkut”, whereas it should be ”kork = fear”. The thing is, ”kork” is emotionally very strong and if we do not get the correct stem, we lose the strong emotion in the sentence. For this reason, we applied the exceptional case for these words, and corrected the stem as ”kork”.

In the preprocessing phase, each word is stemmed, the stop words are removed, and special keywords are added if necessary. Application of these steps on a sample sentence is given in Table 4.15. To emphasize the important points, the word ”kız” detected to be noun and converted to ”kızN”. Since ”bir”, ”biri” and ”bu” are included our stop words list, these words are removed from the sentence at the stop word removal process. Negation is detected in the word ”dayana**ma**dı” and it is converted to ”_dayan”. At last, special suffixes are detected and keyword ”KUC” is added for ”kız**cağız**”, and YT is added for the ability suffix in ”dayan**a**madı”.

4.3 Feature Selection and Feature Vector Construction

After preprocessing is finished, another important task is the feature selection. Since features are the items that our data will be represented with, selecting good features is crucial. Different approaches are combined to get the most powerful result. At first, all distinct words are considered as features, like in bag of words (BOW) approach. Then n-grams are extracted and Weighted Log Likelihood Ratio (WLLR) [43] scoring is used for generating scores to the features. The number of n-grams, up to trigrams, is given in Table 4.16. For instance, in our ISEAR dataset there exists 3757 distinct words (unigrams), 26314 distinct bigrams and 34234 distinct trigrams. All these possible features are examined by using the method described in the next section.

Table 4.16: N-grams Count for Each Dataset

Dataset	**# unigrams**	**# bigrams**	**# trigrams**
ISEAR	3757	26314	34234
Fairy Tales	1848	7774	7991
Combined	3951	32468	41681

4.3.1 Generating the Feature Scores

By using WLLR, the most distinctive features are extracted and this affects the success of the system considerably, as can be seen in results Chapter 5. Each unigram, bigram, trigram's score is calculated by using the WLLR formula given in Chapter 3, Section 3.1.4.2. Assume that we are calculating the scores of unigram features in class *joy*. Then, for a unigram u in class joy, at first u's appearances in class joy is counted and it is divided by total unigram counts in class joy. Let's call this number x. Then another number is calculated, y, which is, the number of occurrences of u in classes other then joy (sad, anger, fear) divided by the total unigram counts in the classes other then joy. The WLLR of u, is $x \times \log(\frac{x}{y})$. This process is applied for each unigram, bigram and trigram in each class. Then, n-grams are sorted in descending order of scores and top n n-grams are selected in this study. This n value is determined on the basis of the experimental results. Since this selection is done for each class, the most valuable n features are extracted for each class. Most distinctive two sets of unigrams, bigrams and trigams are given in Table 4.17 and Table 4.18, for ISEAR and fairy

Table 4.17: Most Distinctive Features for ISEAR Dataset

Unigrams	**Bigrams**	**Trigrams**	**Class**
mutlu kazan	mutlu ol çok mutlu	çok mutlu ol kabul et öğren	joy
sinirlen arkadaş	çok sinirlen çok kız	zaman çok sinirlen suç _ol hal	anger
kork gece	çok kork ol kork	kork film izle diye çok kork	fear
üzül vefat	çok üzül vefat et	çok üzgün hisset zaman çok üzül	sad

tales datasets separately. Top 20 scored unigrams of combined dataset is given in Appendix A.3.

4.3.2 Feature Vector Construction

We have implemented three different feature value assignment methods in this study. The first one is presence - non presence, giving 0 if the feature appears in the sentence, giving 1 if not. This approach did not produce successful results, and this is not unexpected, since the count of the emotional features in the sentence is very important in our case. The other two approaches are more useful for our purpose, so the results are given with those two methods which, are tf and tf-idf weighting. These two approaches are implemented for both of the feature selection methods to see their effects. What we do in this process is generating a matrix of sentence - features. Each sentence is converted to an array of integers, each integer is a value of a feature. If frequency count is the weighting method, the value is the number of occurrences of the feature in the sentence. For example assume that our sentence is ;

Table 4.18: Most Distinctive Features for Fairy Tales Dataset

Unigrams	**Bigrams**	**Trigrams**	**Class**
güzel mutlu	çok sevin çok sev	ömür boyu mutlu YT yerin gel	joy
sinirlen bağır	bağır bağla çok sinirlen	için çok öfke anlam YT _ver	anger
kork korkunç	çok kork kork içeri	korkunç ses bağır merdiven altın geç	fear
ağla üzül	çok üzül ağla bağla	YT yer düş için çok üzül	sad

{*"Çok güzel bir gün geçirdim" dedi bana, ben de onun güzel vakit geçirmesine çok sevindim, içimdeki huzursuzluk kalktı bir anda.*}

After preprocessing it will be seen as;

{ *çok güzel gün geçir dedi güzel vakit geç çok sevin iç _huzur kalk an* }

And suppose that our features are selected as { *güzel, sevin, kız* } . Then, with frequency count weighting approach, the array of this sentence will be, [2, 1, 0].

Tf-idf weighting aims to discriminate more important terms than the others. In this study, we calculated each feature's idf, by the Equation 3.10b. If the sentence contains the feature f, then the weight of the feature is term frequency count multiplied by idf of that feature. In our example above, suppose our document contains 2 sentences:

"Çok güzel bir gün geçirdim" dedi bana, ben de onun güzel vakit geçirmesine çok sevindim, içimdeki huzursuzluk kalktı bir anda.

Bana hediye almış, çok sevindim.

And features are again { *güzel, sevin, kız* } . Then df of our features are [1, 2, 0], and for instance idf of the first feature is calculated with { $\log \frac{2}{1}$ } . Then for the first sentence, tf-idf values are;

$[2 \times \log \frac{2}{1}, 1 \times \log \frac{2}{2}, 0 \times \log \frac{2}{0}]$.

After generating array of feature values for each sentence, data is ready for classification.

4.4 Classification

Main focus in this study is emotion classification and several methods are used and compared for this purpose. Our first dataset, ISEAR dataset, has four basic emotions, *joy*, *sad*, *fear* and *anger*. Second dataset is Turkish fairy tales dataset and here, we have 5 classes, one more than from ISEAR dataset which is ”none”. Also we have the annotated data, labeled with 3 main levels for each emotion, high, medium, low in the fairy tales dataset. At first, classification is applied to the two datasets separately, then the datasets are combined for four emotion categories. The classifications that are applied are;

- ISEAR dataset, with 4 emotion classes
- Fairy tales dataset, with 5 emotion classes
- Combined dataset, with 4 emotion classes
- Fairy tales dataset for each class, with 3 levels.

Classification with these classes are tried with different methods and several tests are applied. The effects of the following items are examined in the study and the best results obtained under the optimal combination of the below steps are presented in Chapter 5 Experiment Results.

- Using different classification methods, naive bayes classifier, complement naive bayes and SVM.

- Using bigrams and trigrams in addition to unigrams
- Using WLLR scoring in feature selection
- Using tf or tf-idf weighting in feature vector construction.

As stated before, Weka tool is used to get the results in this process, since it gives the opportunity of applying different classification methods. In Naive Bayes Classifier, classification is based on the probability distribution function given in Equation 3.2. In default mode, to approximate $P(x|c_i)$(class conditional probability), in Equation 3.2, the assumption is that, the distribution of data follows normal distribution. Weka provides the option of using a kernel estimator rather then a normal distribution. In kernel density estimator, instead of assuming normal distribution, any distribution can be approximated. Using this option increases the computing time and space, but gives better results.

In Complement Naive Bayes, the algorithm does not approximate the probability of x given c, but approximates the probability of x given complement of c. There exists a parameter in Weka for CNB, which is the smoothing variable. For a word that is not seen in the training set, not to have a zero value for the probability, the smoothing variable is used. We used the default parameter, which is 1, in this study.

As explained before, LibSVM depends on a decision function which determines the position of the item in the space. The decision function of LibSVM is given in, Chapter 3. Decision surface can be in many shapes and Equation 3.8 is reconstructed for arbitrary types of decision surfaces, as;

$$I(x) = \sum_{i=1}^{l} \alpha_i y_i K(x, x_i) + b \tag{4.2}$$

where $K(x, x_i)$ is the kernel function. Linear, polynomial and radial basis (RB) are the basic types of kernel functions. The choice of kernel is a very important decision, since it affects the process a lot. RBF kernel is the one that is generally the first choice. SVM has many parameters to be given and especially important ones are the cost and gamma values. They have also high importance, that they make considerable difference on the results. Parameter C, is the penalty parameter, it affects the trade off between complexity of the decision function and errors of the training data. Gamma is the value that controls the shape of the separating hyperplane. Good selection of these values may make the process very successful, whereas poor selection may cause the system to fail. LibSVM provides a script to extract the best cost

and gamma values, ”grid.py”. This script is run with our data, and the best values are given to the system. Linear kernel is another kernel type, and is more efficient than RBF kernel. In our system, linear kernel resulted slightly better than RBF kernel, and since it is more efficient, it is decided to be used in the experiments.

CHAPTER 5

EXPERIMENTAL RESULTS

In this chapter, details of the experimental settings and experiment results are presented. Experiments are performed on three sets of data which are ISEAR dataset, Turkish fairy tales dataset and the combined dataset. At first, classification of emotional categories are applied and then the experiments for the classification of the emotional levels are conducted on Fairy Tales dataset. The effects of using different methods are analysed and compared to each other.

The evaluation of these systems are based on some measurement terms regarding the success of the results. These are accuracy, F-measure, Kappa value, precision and recall. Accuracy is the one that we take into consideration more than others. It is simply the percentage of correctly classified items over all items. Kappa is a chance-corrected measure of agreement between the result of the system and the true classes. As it is explained in the annotation agreement, kappa considers the probability of correctly classifying the items just by chance, and builds it's measurement with that fact. F measure calculation is based on precision and recall values. Precision is the number of correctly predicted items of a class divided by the number of total items that are predicted to be in that class. Recall represents the concept of how much of the items of a class is correctly predicted. F measure is the harmonic mean of precision and recall and calculated with;

$$F = 2 \times \frac{precision \times recall}{precision + recall} \tag{5.1}$$

Higher values of these measures indicate more successful results and different classifiers and features sets results are compared on the basis of these measures. Classifications are validated by using 10 fold cross validation policy, and results for each data set is given in each section.

5.1 Classification Results of ISEAR Dataset

The aim is to classify data to four basic emotions, joy, sad, anger and fear. The experimental results are shown below with different classifiers separately. Dataset contains 4265 samples in total, 1073 from 'joy', 1036 from 'sad', 1083 from 'anger' and 1073 from 'fear' class.

At first, classification is performed by using all distinct words as features under tf and tf-idf weighting methods for feature value assignment. After that, rather than using all distinct word as features, we used WLLR ranking in the feature selection phase, and perform the experiments with different combinations of n-grams. It should be noted that the features with high scores are selected equally for each class.

Experiments with Naive Bayes Classifier

Table 5.1: Classification Results of ISEAR Dataset with NB Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods

		Accuracy	Kappa	Mean F
Total = 3757 features (All distinct words)	**tf**	71.50%	0.62	0.72
	tf-idf	60.60%	0.47	0.61
Total = 800 features **(100 unigram + 100 bigram from each class by WLLR)**	**tf**	63.91%	0.52	0.65
	tf-idf	58.19%	0.44	0.59
Total = 1600 features **(200 unigram + 100 bigram +100 trigram from each class by WLLR)**	**tf**	76.74%	0.69	0.77
	tf-idf	69.33%	0.59	0.69
Total = 2400 features **(200 unigram + 200 bigram +200 trigram from each class by WLLR)**	**tf**	**78.12**%	0.71	0.78
	tf-idf	71.65 %	0.62	0.72
Total = 2400 features **(300 unigram + 200 bigram +100 trigram from each class by WLLR)**	**tf**	78.00%	0.71	0.78
	tf-idf	71.18%	0.61	0.71

As presented in Table 5.1, we can say that the most successful result of NB classifier is the one with 2400 features (200 unigram, 200 bigram and 200 trigram from each class) selected by using WLLR scoring and tf weighting method and at most, 78.12% of accuracy is reached. It is obvious that using tf weighting resulted far more better then tf-idf weighting in all experiments with Naive Bayes classifier. The increasing number of features with WLLR scoring resulted higher accuracy up to a limit, and then the accuracy started to decrease.

Experiments with Complement Naive Bayes Classifier

Table 5.2: Classification Results of ISEAR Dataset with CNB Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods

		Accuracy	Kappa	Mean F
Total = 3757 features (All distinct words)	**tf**	74.70%	0.66	0.75
	tf-idf	72.54%	0.63	0.72
Total = 800 features **(100 unigram + 100 bigram from each class by WLLR)**	**tf**	61.57%	0.49	0.62
	tf-idf	61.69%	0.49	0.62
Total = 1600 features **(200 unigram + 100 bigram +100 trigram from each class by WLLR)**	**tf**	78.97%	0.72	0.79
	tf-idf	78.69 %	0.72	0.79
Total = 2400 features **(200 unigram + 200 bigram +200 trigram from each class by WLLR)**	**tf**	** **81.34%** **	0.75	0.81
	tf-idf	80.89%	0.74	0.81
Total = 2400 features **(300 unigram + 200 bigram +100 trigram from each class by WLLR)**	**tf**	81.27%	0.75	0.81
	tf-idf	80.73%	0.74	0.81

The most successful result of Complement Naive Bayes classifier is the one with 2400 features(200 unigram, 200 bigram and 200 trigram from each class by WLLR) and tf weighting as can be seen from the Table 5.2. The highest accuracy reached is 81.34%, which is a highly encouraging result. We can not say tf or tf-idf weighting resulted better then the other, because the results are mixed. Also, the effect of using different weighting methods did not make a

big difference as did in NB classifier.

Experiments with SVM Classifier

Table 5.3: Classification Results of ISEAR Dataset with SVM Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods

		Accuracy	Kappa	Mean F
Total = 3757 features (All distinct words)	**tf**	72.54%	0.63	0.72
	tf-idf	73.13%	0.64	0.73
Total = 800 features **(100 unigram + 100 bigram from each class by WLLR)**	**tf**	61.01%	0.48	0.76
	tf-idf	60.05%	0.47	0.61
Total = 1600 features **(200 unigram + 100 bigram +100 trigram from each class by WLLR)**	**tf**	74.47%	0.66	0.75
	tf-idf	74.63%	0.67	0.78
Total = 2400 features **(200 unigram + 200 bigram +200 trigram from each class by WLLR)**	**tf**	**75.87**%	0.68	0.76
	tf-idf	75.64%	0.68	0.76
Total = 2400 features **(300 unigram + 200 bigram +100 trigram from each class by WLLR)**	**tf**	75.19%	0.67	0.75
	tf-idf	75.57%	0.67	0.76

The highest result of SVM classifier is reached with the same feature set of NB and CNB Classifiers, 2400 features in total, 200 unigrams, 200 bigrams and 200 trigrams selected with WLLR scoring from each class and tf weighting as given in Table 5.3. The highest accuracy is 75.87% and difference between tf and tf-idf scoring is not very distinctive as in CNB classifier.

After collecting all the results, the highest accuracies reached by each classifier are; NB-78.12%, CNB-81.34% and SVM-75.87%. CNB is the most successful classifier among the others. 81.3365% is a good result on the purpose of emotion classification with 4 classes. The effect of using WLLR in the feature selection process is especially important since it increases success. Using all distinct words may be thought to get the most successful result,

but it is proved that using just the important and valuable features is more effective, and WLLR scoring is very good at selecting the distinctive features.

In Table 5.4, the detailed result of the experiment that gives highest accuracy is shown. The confusion matrix and the accuracy of each class can be seen from the table, the rows are the true classes and the columns are the classification results.

Table 5.4: Confusion Matrix of ISEAR Dataset

CNB Classifier Confusion Matrix					
Class	Joy	Anger	Sad	Fear	**Accuracy**
Joy	915	68	51	39	0.85
Anger	70	881	48	84	0.81
Sad	83	111	780	62	0.75
Fear	53	67	60	893	0.83

Highest accuracy is seen in class 'joy', and the lowest accuracy is seen in class 'sad'. Maximum confusion occurs between 'anger' and 'sad' classes, that 111 of the items that are in 'sad' class, classified to be in 'anger' class erroneously. This is not unexpected when we look at the dataset closer. The writings of people on the incident that they experienced sadness and anger are similar. In some situations, a fact makes someone angry, and the same fact makes the other one sad. This is the reason of confusion between anger and sad classes.

5.2 Classification Results of Turkish Fairy Tales Dataset

In fairy tales dataset, we have 5 classes, joy, sad, anger, fear and none. Four emotional classes, classes except 'none', includes 3 level of emotional states. In this section, at first, classification results of 5 classes are given, separately for each classifier as given with ISEAR dataset. Then results of each class for its emotional state level classifications are shown. The three levels are mainly high, medium and low.

In another set of experiments, we defined two sets of categories as emotional and non-emotional. Four emotional classes are included in "emotional" category and "none" class is included in "non-emotional" category. Experimental results on the classification of two categories are also given.

5.2.1 Results With Emotion Classes

Experiments with Naive Bayes Classifier

Table 5.5: Classification Results of Fairy Tales Dataset with NB Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods

		Accuracy	Kappa	Mean F
Total = 1848 features (All distinct words)	**tf**	63.31%	0.49	0.62
	tf-idf	54.52%	0.35	0.51
Total = 800 features **(100 unigram + 100 bigram from each class by WLLR)**	**tf**	**74.50**%	0.66	0.77
	tf-idf	69.77%	0.57	0.68
Total = 1600 features **(200 unigram + 100 bigram +100 trigram from each class by WLLR)**	**tf**	72.52%	0.63	0.74
	tf-idf	65.12%	0.50	0.62
Total = 2400 features **(200 unigram + 200 bigram +200 trigram from each class by WLLR)**	**tf**	67.96%	0.58	0.71
	tf-idf	65.55 %	0.51	0.62
Total = 2400 features **(300 unigram + 200 bigram +100 trigram from each class by WLLR)**	**tf**	63.65 %	0.51	0.65
	tf-idf	57.97 %	0.39	0.54

Fairy tales dataset results in Table 5.5 with NB classifier shows that the most successful result occurs with 800 features(100 unigram + 100 bigram from each class). The accuracy reached is 74.50% with 5 classes. Increasing number of features decreased the accuracy, and as we have seen with ISEAR dataset, using tf-idf weighting decreases the success of the system.

Experiments with Complement Naive Bayes Classifier

Table 5.6: Classification Results of Fairy Tales Dataset with CNB Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods

		Accuracy	Kappa	Mean F
Total = 1848 features (All distinct words)	**tf**	68.82%	0.5768	0.681
	tf-idf	66.06%	0.54	0.66
Total = 800 features **(100 unigram + 100 bigram from each class by WLLR)**	**tf**	76.83%	0.68	0.74
	tf-idf	**76.83**%	0.68	0.74
Total = 1600 features **(200 unigram + 100 bigram +100 trigram from each class by WLLR)**	**tf**	76.14%	0.67	0.73
	tf-idf	74.59 %	0.65	0.73
Total = 2400 features **(200 unigram + 200 bigram +200 trigram from each class by WLLR)**	**tf**	75.28%	0.66	0.73
	tf-idf	73.38%	0.63	0.72
Total = 2400 features **(300 unigram + 200 bigram +100 trigram from each class by WLLR)**	**tf**	69.51 %	0.58	0.69
	tf-idf	68.30%	0.57	0.69

Using CNB classifier resulted at most 76.83% accuracy, with 800 features as given in Table 5.6. Accuracy of tf and tf-idf weighting seems to be the same, however kappa and f measures are slightly different. When tf weighting is used kappa value is 0.6778, f value is 0.736; however when tf-idf is used kappa value is 0.6787, f value is 0.742. Since kappa and f measures with tf-idf weighting is slightly higher then tf weighting, it is true to say that the most successful result is obtained by using 800 features and tf-idf weighting. A generalization can not be formed on the different results of weighting methods, because the results are mixed.

Experiments with SVM Classifier

Table 5.7: Classification Results of Fairy Tales Dataset with SVM Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods

		Accuracy	Kappa	Mean F
Total = 1848 features (All distinct words)	**tf**	63.39%	0.50	0.63
	tf-idf	63.65 %	0.50	0.63
Total = 800 features **(100 unigram + 100 bigram from each class by WLLR)**	**tf**	**69.42**%	0.59	0.71
	tf-idf	67.70%	0.55	0.66
Total = 1600 features **(200 unigram + 100 bigram +100 trigram from each class by WLLR)**	**tf**	65.72%	0.54	0.67
	tf-idf	66.49%	0.54	0.67
Total = 2400 features **(200 unigram + 200 bigram +200 trigram from each class by WLLR)**	**tf**	65.29%	0.53	0.67
	tf-idf	66.06%	0.54	0.67
Total = 2400 features **(300 unigram + 200 bigram +100 trigram from each class by WLLR)**	**tf**	62.45%	0.48	0.62
	tf-idf	60.55%	0.45	0.60

SVM also concluded the same results as other classifiers. The highest accuracy is reached is 69.42% with 800 features as given in Table 5.7. Again, the weighting methods end with mixed results, one can not say that one is better than the other.

Three different classifiers are tried on fairy tales dataset, and the most successful among the others is CNB classifier. It reached 76.83% accuracy, with 1162 items dataset and 5 distinct classes, using 10 fold cross validation. The most successful result of ISEAR dataset was the one with 2400 features, whereas fairy tales dataset reached the best one with 800 features. At first, it may look weird, however it is a reasonable conclusion. Fairy tales is a small dataset compared to ISEAR, and including many features cause to have many uninformative features, rather then the informative ones. That means, the 800 features are the distinctive ones, when we try to add more features, the added ones are some useless features. And cause the system

to be unsuccessful.

Confusion matrix of the result of CNB classifier with 800 features is as in Table 5.8.

Table 5.8: Confusion Matrix of Fairy Tales Dataset

CNB Classifier Confusion Matrix						
Class	Joy	Anger	Sad	Fear	None	**Accuracy**
Joy	381	6	16	1	3	0.94
Anger	15	114	6	5	9	0.77
Sad	53	8	292	7	8	0.79
Fear	5	7	2	100	10	0.81
None	51	12	32	13	4	0.04

Highest accuracy is seen in class 'joy', and the lowest accuracy is seen in class 'none'. The accuracy of class 'none' is so low, just 4 of the 112 items from 'none' class classified correctly. Nearly, half of the 'none' class items are classified into the class 'joy' erroneously. Some statements may not indicate a feeling explicitly, or may just tell about a fact rather then expressing an emotion. However, the fact that the statement tells, may reveal some joyful feeling in some people, like the sentence "bugün alışverişe çıktım". The reason of the confusion between 'joy' and 'none' classes is this fact.

The other experiment is, dividing dataset into two parts as emotional and non-emotional and classifying these two categories. Four emotional classes constitute the 'emotional' category(1049 items) and 'none' class forms 'non-emotional' class (113 items), which is an unbalanced data. It is explained before, that NB classifier tends to predict the class that has more training samples then the others. Therefore, NB classifier gives the highest accuracy as seen from Table 5.9. However, this is not very informative, if we had more samples from 'none' class, the accuracy would be lower.

Table 5.9: Classification Results of Two Categories with Different Classifiers

Classifier	Accuracy
CNB	82.60%
NB	89.84%
SVM	88.20%

5.2.2 Results With Emotional Levels

The results of three emotional levels are given in this section. CNB classifier is the one that is the most successful and the experimental results with CNB classifier are given in Table 5.10.

Table 5.10: Accuracies of Emotion Intensity Levels

Class	High	Medium	Low	Average
Joy	73.3%	25.2%	17.1%	46.08%
Sad	66.5%	16.1%	30.8%	44.02%
Anger	47.4%	36.5%	57.9%	44.97%
Fear	66.7%	11.1%	22.7%	42.74%

On the average, the accuracies are ranged between 42.74% and 46.08%. It is hard to decide on the level of the statements. The Kappa values of the annotation agreements, given before, also showed that the decision is highly subjective.

5.3 Classification Results of Combined Dataset

Combined dataset contains 5314 items from 4 emotion classes. The same experiments are applied on the combined data to see the effect of mixing two different sources.

Experiments with Naive Bayes Classifier

Table 5.11: Classification Results of Combined Dataset with NB Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods

		Accuracy	Kappa	Mean F
Total = 3951 features (All distinct words)	**tf**	71.56%	0.62	0.71
	tf-idf	58.79%	0.45	0.58
Total = 800 features **(100 unigram + 100 bigram from each class by WLLR)**	**tf**	73.62%	0.65	0.73
	tf-idf	67.56%	0.57	0.67
Total = 1600 features **(200 unigram + 100 bigram +100 trigram from each class by WLLR)**	**tf**	75.37 %	0.67	0.75
	tf-idf	69.27%	0.59	0.69
Total = 2400 features **(200 unigram + 200 bigram +200 trigram from each class by WLLR)**	**tf**	76.25%	0.68	0.76
	tf-idf	70.77%	0.61	0.71
Total = 2400 features **(300 unigram + 200 bigram +100 trigram from each class by WLLR)**	**tf**	**76.66**%	0.69	0.77
	tf-idf	70.64%	0.61	0.70

NB classification results, given in Table 5.11, show that the highest accuracy reached with 2400 features (300 unigram, 200 bigram, 100 trigram from each class) and tf weighting. tf-idf weighting is showed to decrease the success substantially. Higher number of features, that are selected with WLLR scoring improved the system success highly.

Experiments with Complement Naive Bayes Classifier

Table 5.12: Classification Results of Combined Dataset with CNB Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods

		Accuracy	Kappa	Mean F
Total = 3951 features (All distinct words)	**tf**	75.08%	0.67	0.75
	tf-idf	72.34%	0.63	0.72
Total = 800 features **(100 unigram + 100 bigram from each class by WLLR)**	**tf**	76.12%	0.49	0.62
	tf-idf	76.89%	0.69	0.77
Total = 1600 features **(200 unigram + 100 bigram +100 trigram from each class by WLLR)**	**tf**	78.38%	0.71	0.78
	tf-idf	78.02%	0.71	0.78
Total = 2400 features **(200 unigram + 200 bigram +200 trigram from each class by WLLR)**	**tf**	80.07%	0.73	0.80
	tf-idf	80.15%	0.73	0.80
Total = 2400 features **(300 unigram + 200 bigram +100 trigram from each class by WLLR)**	**tf**	**80.39**%	0.74	0.80
	tf-idf	80.09%	0.73	0.80

Results given in Table 5.12, shows the highest accuracy with the same feature set of NB classifier results. The best accuracy is 80.39% and obtained with 2400 features (300 unigrams, 200 bigrams, 100 trigrams) and tf weighting. Difference between the different weighting methods is so slight, that it is less then 1% in general. The highest accuracy of ISEAR dataset was 81.34%, and it seems to be decreased when combined with fairy tales data.

Experiments with SVM Classifier

Table 5.13: Classification Results of Combined Dataset with SVM Classifier Under 10 Fold Cross Validation Using Different Feature Sets and Weighting Methods

		Accuracy	Kappa	Mean F
Total = 3951 features (All distinct words)	**tf**	71.9232%	0.6242	0.719
	tf-idf	72.00%	0.62	0.72
Total = 800 features **(100 unigram + 100 bigram from each class by WLLR)**	**tf**	74.58%	0.66	0.75
	tf-idf	74.82%	0.66	0.75
Total = 1600 features **(200 unigram + 100 bigram +100 trigram from each class by WLLR)**	**tf**	74.12%	0.65	0.74
	tf-idf	**74.95**%	0.66	0.75
Total = 2400 features **(200 unigram + 200 bigram +200 trigram from each class by WLLR)**	**tf**	74.56%	0.66	0.75
	tf-idf	**74.95**%	0.66	0.75
Total = 2400 features **(300 unigram + 200 bigram +100 trigram from each class by WLLR)**	**tf**	74.58%	0.66	0.75
	tf-idf	74.86%	0.66	0.75

This classifier resulted in a different way then the other classifiers. As can be seen in Table 5.13, highest accuracy is 74.95% and reached with two feature sets 2400 features (200 unigrams, 200 bigrams and 200 trigrams from each class) and 1600 fetuares. It is also obvious that, the results of this classifier with distinct feature sets are too close to each other. All the WLLR scored features gave accuracy around 74%. The other two classifiers obtained the most successful result with a different feature set, however the different features did not affect the accuracy with this classifier.

Confusion matrix of the best resulted classifier and feature set is given in Table 5.14. As in other datasets, the highest accuracy is seen in 'joy' class and the highest confusion is between the class 'sad' and 'joy'. With the same experiment set, percentage accuracy versus feature

Table 5.14: Confusion Matrix of Combined Dataset

CNB Classifier Confusion Matrix					
Class	Joy	Anger	Sad	Fear	**Accuracy**
Joy	1264	78	80	59	0.85
Anger	81	975	69	107	0.79
Sad	135	129	1039	101	0.74
Fear	54	80	69	994	0.83

set plotting can be seen in Figure 5.1. Figure 5.1 shows that accuracy is increased up to a point and decreased after that point even if the feature number is increased.

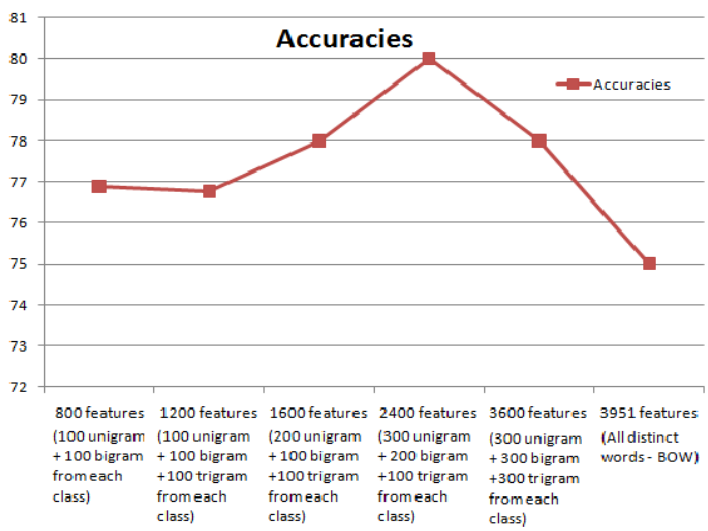

Figure 5.1: Percentage Accuracy versus Feature Set Diagram for Classification with CNB Classifier and Combined Dataset

After applying 10-fold cross validation for each dataset, we have also validated our system by using the same test set for all datasets. The system is trained with each of the training sets and then tested with the same test set separately. Test set includes 1000 items that are collected from ISEAR and Fairy Tales datasets randomly, and testing samples are excluded from the training sets. Tf weighting method is used in the experiments, and the feature sets are selected as the feature sets that have given the best result for each dataset. Accuracy results of our test set are given in Table 5.15. The results of ISEAR and Combined datasets are consistent with 10-fold cross validation results. However, the results with Fairy Tales seem to be lower than the cross validation results, since Fairy Tales dataset is small and weak for testing with 1000

items.

Table 5.15: Accuracies of Test Set with Different Classifiers

Dataset	NB	CNB	SVM
ISEAR	78.41%	80.56%	74.57%
Fairy Tales	64.51%	66.42%	59.32%
Combined	76.95%	79.95%	74.10%

5.4 Evaluation of Overall System and Discussion

Experimental results of three different datasets are given under three different classifiers. When we look at the overall picture, we see that CNB is the most successful classifier among the others, since it gives the best results for all datasets. CNB classifier, learns the weights of the features using the classes except the class it is focused on, and also it overcomes the feature in-dependency assumption of NB. Because of these facts, this type of learning is the one that best fits our data. Since we are using 10-fold cross validation, we have calculated standard deviation of the folds with CNB classifier and the best feature set, to be able to trust our validations. As can bee seen in Table 5.16, 1.671, 3.781 and 1.867 are the standard deviation values for ISEAR, fairy tales and combined datasets respectively.

Table 5.16: Standard Deviations of Cross Validation Results

Dataset	Mean Accuracy	Standard Deviation
ISEAR	81.34%	1.671
Fairy Tales	76.83%	3.781
Combined	80.39%	1.867

It is also shown that increasing the number of features does not always increase the accuracy, extracting the informative and distinctive features is the crucial step. WLLR scoring is proved to get the features that best represent our class distinctions. With smaller number of features, higher accuracies are reached. The number of features that should be used depends on the dataset size. If we have a small dataset, like fairy tales data, having large number of features causes inclusion of uninformative features and hence decreases the accuracy.

The accuracies of high, medium and low levels of emotions are observed to be low. The

importance of annotation agreement is the clue of these low accuracies. By looking at the annotation agreements, we concluded that deciding on the emotional level of statements is hard and personal. If this fact is considered, getting low accuracies is not unexpected.

Confusion matrices show that the highest confusion occurs with the class 'none'. The reason of this is explained earlier and similar to the reason of low accuracies in level classifications. Sometimes, even people may not able to distinguish some statements as emotional and non-emotional. Complicated nature of the task caused the confusion between the class 'none' and the others.

The accuracy result of ISEAR dataset was 81.34% and it is decreased in some degree when the fairy tales dataset is added. However, the combined dataset is supposed to be a more powerful dataset, since it contains a wider area and different styles of texts. It is also expected to provide better understanding of a new statement's feeling.

Furthermore, to test our system with some real life data, we collected some newspaper articles and blog posts from internet and tested that writings with our system. Results are discussed in the next section.

5.5 Experiments With Articles

The articles and blog posts are gathered from the internet and given as testing set to our system. In this set of experiments, since fairy tales dataset is a very small dataset, we do not train the system with just fairy tales data. ISEAR and the combined datasets are used in training, and the results are given separately.

A newspaper article[1] , which has a topic of women exposed to violence, is mainly tells about women's being insulted and despised by some mean people. It states that women's rights abuse is a highly important subject and precautions for these incidents must be taken urgently. When this article is used as test data, both ISEAR and combined training sets give the same answer. The article is classified to be in 'anger' class. The article contains many keywords that give the feeling of anger, such as, violence, intrusion, insulting, etc., and classification result is acceptable.

[1] http://www.tumkoseyazilari.com/yazar/kaan-ozbek/23-05-2012-korkutan-tablo.html, last accessed 15.07.2012.

Another one[2] is about people died on terrorist attacks. The article tells about the mistakes that have been done during the defense operations of our military services, and how catastrophic results would occur because of the mistakes. The keywords of the article may be given as, enemy, attacks, invasions, our dying soldiers, murderers, etc. Both of the training sets classified this example with the same class, 'fear'. The emotion of the writer himself is not actually fear; however when the statements of the texts are considered we can not say that result is exactly wrong.

The other article[3] is classified in separate classes by different training sets. Article is about the poor lives of public servants that work for the government. The article discusses the payments of these people, the challenges they face, and the problems they encounter because of being lack of affording many products. When ISEAR dataset is used in training, the article is included in 'fear' class, however, combined training set says that the article is in 'sad' class. Prediction of combined dataset is more suitable than the other, since the dominant feeling of the article is sadness.

A different style of writing is obtained from a blog site[4]. The writing is tested; however it does not express an emotion directly. Rather, the feeling of the writing can be described as melancholy. The writer tells about an island's past times, how peaceful and relaxing it was in the past, talks to elder people about their feelings on the island, tells about the winter of the island, and says that the island is the place to be together with just herself, is the right place to trip to inside of herself. This highly melancholic writing is classified in 'fear' class by both training sets. Fear may not be the right choice of emotion, however as it is explained there does not exist a particular emotion in the text.

Another newspaper article[5] is about the deficiency of the justice system and the problems of our being state of law. The incomplete cases, and the free criminals that should be in jail is the main focus of the article. It is stated that people walking on street are in danger because of the thinner addicts, and this is because of the defects of the justice system. When the system is trained with ISEAR dataset, emotion predicted is 'fear'; and when the system is trained with combined dataset, emotion predicted is 'sad'. In the article both emotions have been

[2] http://www.internethaber.com/operasyon-icin-35-sehit-gerekiyormus-meger-12120y.htm, last accessed 15.07.2012.

[3] http://www.ilk-kursun.com/haber/105640, last accessed 15.07.2012.

[4] http://www.mutlulukbizim.com/kargac-k-burgazc-k-bir-guz-sabah, last accessed 15.07.2012.

[5] http://www.yazaroku.com/fyasam-magazin/hincal-uluc/06-01-2010/adaletin-olmadigi-hukuk-devleti/182475/.aspx, last accessed 15.07.2012.

expressed, so both results are acceptable.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, we focused on emotion analysis of Turkish texts and it is shown that using ML methods for Turkish texts on analysis of emotions is feasible and gives promising results. Several methods are applied to get the best results with Turkish language and experimental results are evaluated.

For the purpose of emotion classification, we searched for available datasets that are suitable for our study. Having not found any studies and datasets on this subject, we generated a new dataset. Two types of sources are used while creating the dataset, the first one is an English oriented ISEAR [3] dataset and the second one is Turkish fairy tales. To decide on which emotions are going to be analysed, Ekman's List [2] of emotions, emotions covered in ISEAR dataset and emotions that are dominant in fairy tales are examined. After the examination, four basic emotions are revealed which are joy, sadness, anger and fear.

ISEAR dataset, which is composed of questionnaire answers of many people from different countries and cultures, is translated to Turkish with the help of 33 people. Turkish fairy tales is the second source, at first possible emotional statements are extracted, then the statements are labeled. Annotation agreements are measured with Kappa to be able to comment on the annotations. It is seen that, if the agreement of the annotation is high, then the success of classification is also high. These two data sources are then combined to have a bigger dataset that covers different natures.

In the preprocessing phase, we applied stemming, removed stop words and examined several situations not to cause any information loss. Morphological structure of Turkish is taken into consideration and added many exceptions for Turkish language. Handling negations and some special suffixes is important, since it expands the knowledge and increases the accuracy. In

the feature selection phase, WLLR scoring is tried with combinations of n-grams and it is proved to be an important step that increases the success of the system. From the results it can be said that feature selection with WLLR, helped to increase the success with smaller number of features. The most distinctive words list of our dataset is created and this list is showed to represent the general view of our dataset. Tf and tf-idf weighting methods are applied to see the effects of them; however a generalization on which one is better could not be made, both of them give close results.

Experiments with different classifiers show that CNB classifier is the best one among the others. NB and SVM are also tried, but they could not catch up the results of CNB classifier. Using kernel estimators in NB classifier increases the success and run time at the same time. Among different kernel functions of SVM, linear kernel is the most suitable for our case. None of the improvements of NB and SVM did give better results than CNB.

At first the system is evaluated by using 10 fold cross validation, and then some other writings are used to test our study. ISEAR dataset classification with 4 classes reached 81.34%, fairy tales dataset classification with 5 classes reached 76.83% and combined dataset classification with 4 classes reached 80.39% of accuracies by using CNB classifier. Results of emotion level classification as high, medium and low are also given for the fairy tales dataset. Because level discrimination is a very hard task, accuracies obtained are low. Further experiments are conducted on the newspaper articles and some blog posts. Emotions of the writings are examined and results are encouraging.

Collecting original Turkish writings as the training dataset would be more appropriate rather than translating an English dataset. Such a dataset would also give better results with experiments on articles and blog posts. If writings of people including demographics data, such as age and gender information, can be gathered, a research may be conducted to see the relation between emotion and gender as an instance. Such a study is conducted within a sentiment analysis project and the influence of factors such as age, gender and education level is investigated [54].

This study can be used for observing the general emotion of people on a particular event or subject. To give an example, twitter posts may be analysed to learn the public feeling on an important event and automatic learning is crucial with that giant source of data.

Including more NLP operations may provide some other relevant information. For example, grammatical structure of the statements, such as tense usages may be analysed to see the general convention of different emotions. Dividing the sentence to its components and embedding that information to the process may also be another informative application. Deriving the subject, object, verb dependencies may help to detect the semantics of the statements.

There exists some studies on using a weighted feature support vector machines (WFSVM) [55], rather than the regular one. In the regular approach, each feature has the same effect to the classification; however some features are more important than the others. WFSVM approach tries to give more importance to more valuable features. Adapting this approach may also be applied to the existing system of ours to see the effect of it.

To be able to analyse more emotions, our dataset may be expanded with some other emotions as a future work.

REFERENCES

[1] C.I. Nass, J.S. Stener, and Tanber, *E. Computers are social actors*. In Proceedings of CHI '94, (Boston, MA), pp. 72-78, April 1994.

[2] P. Ekman, M.J. Power (Ed.), *In Handbook of cognition and emotion*, pages 45-60, 1999.

[3] K.R. Scherer and H.G. Wallbott, *Evidence for universality and cultural variation of differential emotion response patterning*. Journal of Personality and Social Psychology, 66:310-328, 1994.

[4] C. Elliott, *The affective reasoner: a process model of emotions in a multi-agent system*, Doctoral thesis, No rthwestern University, Evanston, IL, May 1992 .

[5] J. Olveres, M. Billinghurst, J. Savage, and A. Holden, *Intelligent, expressive avatars*. In Proceedings of the First Workshop on Embodied Conversational Characters, pp. 47-55, 1998.

[6] C. Strapparava and Valitutti, *A. WordNet-Affect: an affective extension of WordNet*. In Proceedings of the International Conference on Language Resources and Evaluation, pp. 1083-1086 2004

[7] A. Ortony, G.L. Clore, and M.A. Foss, *The referential structure of the affective lexicon*. Cognitive Science, 11, 341-364. 1987.

[8] A.C. Boucouvalas, *Real time text-to-emotion engine for expressive Internet communications*. In Being There: Concepts, Effects and Measurement of User Presence in Synthetic Environments, Ios Press, pp. 306-318, 2003.

[9] A. Neviarouskaya, H. Predinger and M. Ishizuka, *Affect Analysis Model: novel rule-based approach to affect sensing from text*. Natural Language Engineering, 17 , pp 95-135 doi:10.1017/S1351324910000239,2011.

[10] M. Al M.Shaikh, *An analytical approach for affect sensing from text*. PhD thesis, University of Tokyo, 2008.

[11] A. Ortony, G. L. Clore, A. Collins, *The Cognitive Structure of Emotions*, Cambridge University Press, 1988.

[12] M. Yashar, *Role of emotion in information retrieval*, University of Glasgow, PhD thesis, School of Computing Science, 2012.

[13] H. Liu, H. Lieberman, T. Selker, *A Model of Textual Affect Sensing using Real-World Knowledge*. Proceedings of the 2003 International Conference on Intelligent User Interfaces, IUI 2003, January 12-15, 2003, Miami, FL, USA. ACM 2003, ISBN 1-58113-586-6, pp. 125-132. Miami, Florida, 2003.

[14] P. Singh, T. Lin, T. Mueller, G. Lim, T. Perkins, W. Zhu, *The Open Mind Common Sense: Knowledge acquisition from the general public*, In On the Move to Meaningful Internet Systems, DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002, Robert Meersman and Zahir Tari, Springer-Verlang, London, UK, 1223-1237, 2002.

[15] C. Alm and D. Roth and R. Sproat, *Emotions from text: machine learning for text-based emotion prediction*. EMNLP, 2005.

[16] C. Strapparava and A. Valitutti, *WordNet-Affect: an affective extension of WordNet*. In Proceedings ofthe 4th International Conference on Language Resources and Evaluation, Lisbon, May 2004, pp. 1083-1086, 2004.

[17] S. Aman, S. Szpakowicz, *Identifying Expressions of Emotion in Text*. Text Speech and Dialogue 4629, 196-205, 2007.

[18] J. Cohen, *A coefficient of agreement for nominal scales*, Educational and Psychological Measurement. 20:37-46, 1960.

[19] S. Aman, S. Szpakowicz, *Using Roget's Thesaurus for Fine-grained Emotion Recognition*. Emotion 312-318, 2008.

[20] C. Strapparava and R. Mihalcea, *Learning to Identify Emotions in Text*. Proceedings of the 2008 ACMSAC'08

[21] P. Katz, M. Singleton, R. Wicentowski, *SWAT-MP: The SemEval-2007 Systems for Task 5 and Task 14*. In Proc. of SemEval-2007, 2007.

[22] C. Strappava and R. Mihalcea, SemEval-2007 Task 14: Affective Text, In: Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 70-74, Prague, June, 2007.

[23] R.A. Calvo and M. Kim, *Emotions in text: dimensional and categorical models*, 2012 - to appear.

[24] T. Danisman and A. Alpkocak, *Feeler: Emotion Classification of Text Using Vector Space Model*. AISB 2008 Convention, Scotland, 2008.

[25] S. Satapathy, S. Bhagwani, *Capturing Emotions in Sentences*, 2011.

[26] U. Eroğul, *Sentiment Analysis in Turkish*, Middle East Technical University, Ms Thesis, Computer Engineering, 2009.

[27] N.B. Albayrak, *Opinion and Sentiment Analysis Using Natural Language Processing Techniques*, Fatih University, Ms Thesis, Computer Engineering, January 2011.

[28] S.İlhan, N. Duru, Ş. Karagöz, M. Sağır, *Metin Madenciliği ile Soru Cevaplama Sistemi*, ELECO 2008, 356-359, 2008.

[29] A. Karadağ, H. Takçı, *Metin Madenciliği ile Benzer Haber Tespiti*, AB 2010, Akademik Bilişim, Muğla Üniversitesi, Muğla, Şubat 2010.

[30] A. Güven, Ö. Bozkurt, O. Kalıpsız, *Gizli Anlambilimsel Dizinleme Yönteminin n-gram Kelimlerle Geliştirilerek İ leri Düzey Dökuman Kümelemesinde Kullanımı*, Bilgisayar Mühendisliği Bölümü, Yıldız Teknik Üniversitesi.

[31] F. Can, S. Kocberber, O. Baglioglu, S. Kardas, H.C. Ocalan, and E. Iyar, *New event detection and topic tracking in Turkish*. J. Am. Soc. Inf. Sci., 61:802-819. doi: 10.1002/asi.21264, 2010.

[32] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, *The WEKA Data Mining Software: An Update*, SIGKDD Explorations, Volume 11, Issue 1., 2009.

[33] Akın, A.A. & Akın, M.D., *Zemberek, an open source nlp framework for Turkic languages. Structure*, 2007. Available at: http://zemberek.googlecode.com/files/zemberek_makale.pdf

[34] I. Pop, *An approach of the Naive Bayes Classifier for the document classification*, General Mathematics, Vol. 14,No.4, pp. 135-138, 2006.

[35] J.D.M. Rennie, L.Shih, J.Teevan, and D.R.Karger, *Tackling the poor assumptions of naive bayes text classification*. ICML2003, pages 616-623, 2003.

[36] C. Cortes and V. Vapnik, *Support-vector networks*. Machine Learning, 20:273-297, November 1995.

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

[38] C. Chang and C. Lin, *LIBSVM : a library for support vector machines*. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm

[39] T. Joachims, *Making large-Scale SVM Learning Practical*. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

[40] J. Fürnkranz, *A Study Using n-gram features for Text Categorization*. Technical Report OEFAI-TR-98-30, Austrian Research Institute for Artificial Intelligence, Vienna, Austria, 1998.

[41] D. Lewis, *Representation and Learning in Information Retrieval*. Technical Report UMCS-1991-093. Department of Computer Science, University of Massachusetts, Amherst, MA, 1992.

[42] D. Mladenic and M.Grobelnik, *Word sequences as features in text learning*. In Proceedings of the 17th Electrotechnical and Computer Science Conference (ERK-98) (pp. 145-148), Ljubljana, Slovenia, 1998.

[43] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, *Text classification from labeled and unlabeled documents using EM*. Machine Learning, 39(2/3): 103-134, 2000.

[44] V. Ng, S. Dasgupta and S.M.N. Arifin, *Examining the Role of Linguistic Knowledge Sources in the Automatic Identification and Classification of Reviews*. Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, 2006.

[45] B. Pang, L. Lee, and S. Vaithyanathan, *Thumbs up? Sentiment classification using machine learning techniques*, In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 79-86, 2002.

[46] C.D. Manning, P. Raghavan and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[47] E.D. Liddy, *Natural Language Processing*. Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc., 2001.

[48] G.G. Chowdhury, *Natural language processing*. Ann. Rev. Info. Sci. Tech., 37: 51-89. doi: 10.1002/aris.1440370103, 2003.

[49] *Lesson40 Issues in NLP* Version 2 CSE IIT, Kharagpu

[50] W.S. Davis, D.C. Yen, *The Information System Consultant's Handbook: Systems Analysis and Design*, 1998.

[51] S.R. Reddy V, D.V.L.N. Somayajulu, A.R. Dani, *Classification of Movie Reviews Using Complemented Naive Bayesian Classifier*, International Journal of Intelligent Computing Research (IJICR), Volume 1, Issue 4, December 2010.

[52] D. Ghazi, D. Inkpen, S. Szpakowicz, *Hierarchical versus Flat Classification of Emotions in Text*, Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, pages 140-146, Los Angeles, California, June 2010.

[53] M. Abdul-Mageed, M.T. Diab, *Subjectivity and Sentiment Annotation of Modern Standard Arabic Newswire*, Proceedings of the Fifth Law Workshop (LAW V), pages 110-118, Portland, Oregon, 23-24 June 2011.

[54] O. Kucuktunc, B.B. Cambazoglu, I. Weber, and H. Ferhatosmanoglu, *A large-scale sentiment analysis for Yahoo! answers*, In Preceedings of the fifth ACM international conference on Web search and data mining (WSDM '12), ACM, New York, NY, USA, 633-642. DOI=10.1145/2124295.2124371, 2012.

[55] K. Wang, X. Wang, Y. Zhong, *A Weighted Feature Support Vector Machines Method for Semantic Image Classification* International Conference on Measuring Technology and Mechatronics Automation, 2010.

Appendix A

LISTS

A.1 Stop Words List

altmış	altı	bana	ben
benden	beni	benim	bey
bin	bir	biri	birini
biz	bizde	bizi	bize
bizden	bizi	bizim	bu
buna	bunda	bunlar	bunları
bunların	bunu	bunun	burada
da	de	doksan	dokuz
dolayısıyla	dört	edecek	eden
ederek	edilecek	ediliyor	edilmesi
ediyor	elli	en	etmesi
etti	ettiği	ettiğini	gibi
hangi	herhangi	iki	ile
ilgili	ise	işte	itibariyle
katrilyon	ki	kim	kimden
kime	kimi	kırk	milyar
milyon	mu	mı	mü
nasıl	ne	nerde	nerede
nereye	o	olan	olarak
olsa	olup	olursa	on
ona	ondan	onlar	onlardan
onları	onların	onu	onun
otuz	öyle	pek	sekiz
seksen	sen	senden	seni
senin	siz	sizden	sizi
sizin	şey	şeyden	şeyi
şeyler	şöyle	şu	şuna
şunda	şundan	şunları	şunu
trilyon	tüm	üç	üzere
var	vardı	ve	veya
ya	yani	yedi	yetmiş
yirmi	yüz	falan	

A.2 Fairy Tales List

Güzel ve Çirkin
Keloğlan ile Vefasız Arkadaşı
Kibritçi Kız
Kücük Deniz Kızı
Kırmızı Başlıklı Kız
Kül Kedisi
Yoksul Oduncu
Sihirli Fasülye
Fareli Köyun Kavalcısı
Altın Saçlı Kız
Şifalı Su
Uyuyan Güzel
Orman Perisinin Gülleri
Yavru Köpek Sevgisi
Padişahın Elbisesi
Çirkin Ördek Yavrusu
Bremen Mızıkacıları
Arslan ile Fare
Şampiyon Ördek
Nilüfer Perisi
Kurbağacık
Kayıp Kasaba
Yalanla Kurulan Dünya
Yeniden Hayata
Bilinmeyen Varlıklar Ailesi

A.3 Most Distinctive Words of Combined DataSet

Rank	Joy	Sad	Anger	Fear
1	mutlu	üzül	sinirlen	kork
2	güzel	vefat	kız	karanlık
3	kazan	üzgin	söyle	gece
4	sevin	zaman	suçla	araba
5	kabul	ayrıl	_ver	ev
6	sevinç	ölüm	sinir	yürü
7	iyi	ağla	_hak	yol
8	sınav	çok	öfke	yılan
9	al	yakın	saçma	hayalet
10	üniversite	arkadaş	kavga	adam
11	geç	YT	konu	yalnız
12	başarı	baba	para	kaya
13	seçil	kanser	aşağıla	takip
14	hediye	kaybet	başka	gir
15	neşe	acı	ama	ıssız
16	ödül	kal	tartışma	tehdit
17	doğ	üzüntü	sor	saat
18	öğren	cenaze	izin	buz
19	teklif	veda	bağır	tek
20	peri	hasta	_yap	kaza