

IRONY DETECTION ON TURKISH MICROBLOG TEXTS

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

BY

HANDE TAŞLIOĞLU

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

SEPTEMBER 2014



Approval of the thesis:

**IRONY DETECTION ON TURKISH MICROBLOG TEXTS**

Submitted by **HANDE TAŞLIOĞLU** 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. Dr. Pınar Karagöz  
Supervisor, **Computer Engineering Department, METU**

\_\_\_\_\_

**Examining Committee Members:**

Prof. Dr. İsmail Hakkı Toroslu  
Computer Engineering Department, METU

\_\_\_\_\_

Assoc. Prof. Dr. Pınar Karagöz  
Computer Engineering Department, METU

\_\_\_\_\_

Assoc. Prof. Dr. Osman Abul  
Computer Engineering Department, TOBB UET

\_\_\_\_\_

Assoc. Prof. Dr. Tolga Can  
Computer Engineering Department, METU

\_\_\_\_\_

Prof. Dr. Ahmet Coşar  
Computer Engineering Department, METU

\_\_\_\_\_

**Date: 05.09.2014**

**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: HANDE TAŞLIOĞLU

Signature :

## ABSTRACT

### IRONY DETECTION ON TURKISH MICROBLOG TEXTS

Taşlıođlu, Hande

M.Sc., Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Pınar Karagöz

September 2014, 72 pages

Social media is the new trend for expressing personal ideas to other people. Since people are sharing real time messages about their opinions on diverse topics, there exists huge amount of raw data to analyze. Thus, manual classification of these data becomes impossible.

Irony, as a simple definition, is creative use of language and attracts computer scientists' attention lately. Automatic detection of irony on microblog texts is not a trivial task. Texts of microblogs can have limited number of characters, mostly include typing errors therefore traditional methods of opinion classification cannot be applied easily. Therefore, a preprocessing requirement is occurred. After preprocessing, some patterns and language specific features are extracted in order to detect irony.

This study aims to automatically detect the irony in microblogs, i.e., informal short texts. Due to the morphological structure of Turkish, various methods are applied to increase the success and quality of classification.

Keywords: Irony Detection, Turkish, Natural Language Processing, Microblogs,  
Sentiment Analysis

## ÖZ

### TÜRKÇE MİKROBLOG YAZILARINDA İRONİ ANLAMA

Taşlıođlu, Hande

Yüksek Lisans, Bilgisayar Mühendisliđi Bölümü

Tez Yöneticisi: Doç. Dr. Pınar Karagöz

Eylül 2014, 72 sayfa

Sosyal medya kişilerin kendi fikirlerini başkalarına aktarmalarındaki yeni akımdır. İnsanların deđişik konulardaki fikirlerini gerçek zamanlı paylaşmasıyla analiz edilecek bir çok ham veri vardır. Ancak, bu verilerin manuel olarak sınıflandırılması neredeyse imkansızdır.

İroni, basit bir tanım ile, dili yaratıcı bir şekilde kullanmaktır ve son zamanlarda bilgisayar bilimcilerin dikkatini çekmektedir. Mikroblog yazılarının limitli karakterlerden oluşması ve çokça yazım hatasının olması nedeniyle, sınıflandırma yaparken geleneksel yöntemler yetersiz kalır. Bu yüzden bu veriler üzerinde ön işleme yapma ihtiyacı doğmuştur. Bu ön işleme yapıldıktan sonra, ironiyi anlayabilmek için dile özgün özellikler ve bir takım kalıplar çıkarılmıştır.

Bu çalışma Türkçe bilgi seti kullanarak sosyal medyadaki öznel cümleleri bulup; onları ironi içerip içermemesi bakımından sınıflandırmayı hedefler. Sınıflandırmanın kalitesini ve başarısını arttırmak için deđişik yöntemler uygulanmıştır.

Anahtar Kelimeler: İroni Anlama, Türkçe, Doğal Dil İşleme, Mikrobloglar, Duygu  
Çözümleme

*To my family.*

## **ACKNOWLEDGEMENTS**

I express my sincere appreciation to my supervisor, Assoc. Prof. Dr. Pınar Karagöz, 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 express my very great appreciation to Alper Sağlam, for his precious support, his endless love and for being the source of my motivation. It would be impossible to finish this work without his support.

I am very grateful and would like to thank my family for their invaluable patience and encouragement.

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

## TABLE OF CONTENTS

ABSTRACT .....	v
ÖZ .....	vii
ACKNOWLEDGEMENTS .....	x
TABLE OF CONTENTS .....	xi
LIST OF TABLES.....	xv
LIST OF FIGURES.....	xvii
CHAPTERS	
1 INTRODUCTION .....	1
1.1 Motivation.....	1
1.2 Morphology of Turkish .....	2
1.3 Outline of the Thesis .....	3
2 BACKGROUND INFORMATION.....	5
2.1 Twitter .....	5
2.1.1 Twitter API.....	6
2.2 Irony .....	6
2.3 SentiStrength.....	7
2.4 Natural Language Processing.....	8
2.4.1 Part of Speech .....	9
2.4.2 Zemberek.....	9
2.5 Machine Learning .....	10
2.5.1 Naïve Bayes Classifier .....	11

2.5.2 Support Vector Machines .....	12
2.5.3 K-Nearest Neighbors.....	13
2.5.4 Random Forests.....	14
3 RELATED WORK.....	17
3.1 Previous Studies on Irony Detection.....	17
3.2 Previous Studies on Sentiment Analysis .....	23
4 METHODOLOGY.....	27
4.1 Data Collection.....	28
4.2 Data Annotating.....	28
4.3 Data Preprocessing.....	30
4.3.1 URL Extraction .....	30
4.3.2 Detection of Hashtags, Usernames and Retweets .....	31
4.3.3 Uppercase Identification .....	31
4.3.4 Detecting Proper Names.....	32
4.3.5 Spell Checking and Correction .....	32
4.4 Feature Extraction .....	33
4.4.1 Bracketed Question Mark / Exclamation Mark.....	34
4.4.2 Exclamation Marks .....	34
4.4.3 Question Marks .....	35
4.4.4 Quotation Marks.....	35
4.4.5 Full Stop (Ellipsis) .....	36
4.4.6 Smileys.....	36
4.4.7 Frowns.....	36
4.4.8 Gap between Sentiment Scores .....	37
4.4.9 Diminutive Forms .....	38

4.4.10 Interjections.....	38
4.4.11 Upper Case .....	39
4.5 Classification.....	39
5 ANALYSIS .....	41
5.1 Evaluation Metrics .....	42
5.2 Analysis on Classification Algorithms for Irony Detection .....	42
5.2.1 Experimental Results under Naïve Bayes Classifier.....	43
5.2.2 Experimental Results under Support Vector Machine Classifier .....	44
5.2.3 Experimental Results under k-Nearest Neighbor Classifier.....	45
5.2.4 Experimental Results under Random Forests Classifier .....	46
5.2.5 General View on the Performance of Classification .....	47
5.3 Analysis on the Effect of Attributes on Irony Detection .....	49
5.3.1 Experimental Results under Best First Attribute Selection Technique .....	50
5.3.2 Experimental Results under Ranker Attribute Selection Technique .....	51
5.4 General Overview of Effects of Attributes on Irony Detection .....	52
5.5 Effects of Model with Annotated Data.....	55
5.6 Analysis for Irony Detection with English Dataset.....	56
5.7 Case Study: Analysis on the Effect of Long Texts on Irony Detection.. ..	57
6 CONCLUSION .....	61
REFERENCES.....	63
APPEDICIES	
A LIST OF SMILEYS & FROWNS.....	67

B LIST OF TURKISH INTERJECTIONS .....	69
C SENTENCES AND CLASSIFICATION RESULTS .....	71

## LIST OF TABLES

Table 1 SentiStrength supporting file list.....	8
Table 2 The classification results of annotated data.....	29
Table 3 Spell checking with Zemberek (Example 1).....	32
Table 4 Spell checking with Zemberek (Example 2).....	33
Table 5 Classification results of Naïve Bayes with different datasets .....	43
Table 6 The results of Naïve Bayes Algorithm with different datasets.....	43
Table 7 Classification results of Support Vector Machines with different datasets .....	44
Table 8 The results of Support Vector Machines Algorithm with different datasets .....	45
Table 9 Classification results of k-Nearest Neighbors with different datasets .....	45
Table 10 The results of k-Nearest Neighbors Algorithm with different datasets.....	46
Table 11 Classification results of Random Forests with different datasets.....	46
Table 12 The results of Random Forests Algorithm with different datasets.....	47
Table 13 Classification results with Top 3 attributes (Training set).....	50
Table 14 Classification results with Top 3 attributes (Test set).....	50
Table 15 Classification results of Top 5 attributes (Training set) .....	51
Table 16 Classification results of Top 5 attributes (Test set) .....	52
Table 17 The classification results on annotated data.....	55

Table 18 The classification results of training data used as test data in English.....56

Table 19 The classification results of newly introduced test data in English.....57

Table 20 The classification results for case study .....58

## LIST OF FIGURES

Figure 1 An example of splitting word “bulamaz” .....	10
Figure 2 Correction of mistyped word “önrne” .....	10
Figure 3 The change of hyperplane when classifying with SVM.....	12
Figure 4 The flow of methodology .....	27
Figure 5 The sentiment distribution of irony data.....	30
Figure 6 General overview of the results generated with training set is used for test set scenario .....	48
Figure 7 General overview of the results generated with newly introduced test set scenario .....	49
Figure 8 General overview of results of attribute selection when training set used as test set .....	52
Figure 9 General overview of results of attribute selection with test set .....	54



# CHAPTER 1

## INTRODUCTION

### 1.1 Motivation

People always have curiosity about what other people think about themselves, popular news, etc., in short everything. The rise of the social media in last decade changed the perspective of expressing ideas. Ever after people started to share their opinions in web via social media sources, they start to satisfy their curiosity via web.

Nowadays, the new trend of expressing ideas is Twitter, which is the most popular microblogging website. Users can share their opinions in text based format, limited in 140 characters, in real time. It has started in 2006 and gained worldwide popularity shortly. Twitter has about 645 million registered users by the end of 2013 and they post approximately 58 million tweets daily<sup>1</sup>. Therefore, this situation means that social media sources become a huge database for sentiment analysis.

The data in Twitter is valuable because people use Twitter for sharing ideas about opinions, news, moods. This raw data is a good source for analyzing sentimentally. However, it is hard to do sentimental classification manually, without tools. Sentiment analysis is a part of this study. The results of sentiment analysis are used for irony detection. Because, ironic sentences mostly have positive sentiment score but they have negative meaning at all.

---

<sup>1</sup> <http://www.statisticbrain.com/twitter-statistics/>

Irony detection attracts computer scientists' attention lately. Irony in spoken discourse can be understood from emphasis on some specific words thorough the sentence. It can be easily detected if someone is making irony when talking. However, automatic detection of irony in written discourse is not a trivial task. Some specific clues are seek in order to understand whether the sentence contains irony or not. The writer does not have to include the clues to the text. Irony can be done by saying opposite of the obvious facts without any sign. This type of irony is not in the scope of this study.

The studies conducted in this area are very limited. However, the languages used in these studies are very diverse. There are studies in English, Dutch and Portuguese ... etc. some of the techniques used for detecting irony are common for all languages, but some language specific features are also used for detecting irony.

## **1.2 Morphology of Turkish**

Turkish is the one of the agglutinated languages that makes it to harder to analyze morphologically. Agglutination means that complex words are formed by stringing morphemes which has single grammatical or semantically meaning<sup>2</sup>. This means that words in Turkish can be constructed as adding suffixes to the roots of the words.

Since theoretically new words can be generated by agglutinated structure of Turkish. To give an example, the word “gelmemeliydim”, also a sentence itself, can be expressed in English like “I should have not come”. Therefore, in order to solve this complex structure of the words in Turkish, a morphological analyzer is needed.

In English language, negativity is mostly given to a sentence with negation word “not”. However, in our language, the affixes “-me,-ma” which are attached to verbs are used for negating the sentence. The word “değil” is also used for the same purpose but not as common as “not”. Positivity or negativity may be expressed with adjectives also. The most

---

<sup>1</sup> <http://en.wikipedia.org/wiki/Agglutination>

common approach in English to make sentiment analysis is to get rid of the suffixes. However, this system is not applicable in Turkish, since the reasons explained above.

### **1.3 Outline of the Thesis**

Chapter 2 gives some background information about both irony detection and sentiment analysis. The algorithms, tools and techniques are explained in this part. Chapter 3 mentions about previous studies which are conducted in irony detection. The used methods and how they are constructed are explained in Chapter 4. The results of the analysis are examined in detail in Chapter 5. Lastly, Chapter 6 mentions about conclusion and feature works.



## CHAPTER 2

### BACKGROUND INFORMATION

#### 2.1 Twitter

Twitter is one of the most popular social networking website that was created in 2006. Users can send text messages in Twitter, which are called “tweets” and limited to 140 characters. In tweets, there can be links, ideas, smileys, in short what users want to share.

Registered user profile of Twitter contains every type of personalities such as celebrities, politicians, students, housewives ... etc. This diversity on personalities results in the data that exists in Twitter to include different opinions.

Since Twitter is accessible via web, short messaging service (SMS) and mobile application, the number of tweets is increasing day by day. According to a study, as of September 2013, the company's data showed that 200 million users send over 400 million tweets daily, with nearly 60% of tweets sent from mobile devices<sup>3</sup>.

Some terms that are specific to Twitter need to be clarified. Firstly, hashtag is simply a word with hash character (#) appended to the beginning. In Twitter, tweets with hashtag can be queried and user can get the set of the tweets that contain this hashtag. Secondly, retweet means sharing someone’s tweet.

---

<sup>3</sup> [http://www.theguardian.com/technology/2013/sep/12/twitter-ipo-stock-market-launch?CMP=EMCNEWEML6619I2&et\\_cid=48826&et\\_rid=7107573&Linkid=http%3a%2f%2fwww.theguardian.com%2ftechnology%2f2013%2fsep%2f12%2ftwitter-ipo-stock-market-launch](http://www.theguardian.com/technology/2013/sep/12/twitter-ipo-stock-market-launch?CMP=EMCNEWEML6619I2&et_cid=48826&et_rid=7107573&Linkid=http%3a%2f%2fwww.theguardian.com%2ftechnology%2f2013%2fsep%2f12%2ftwitter-ipo-stock-market-launch)

### 2.1.1 Twitter API

Twitter exposes its data via Twitter API. Developers can get tweets and some other information like friends list, tweet date, users' personal information.

There are two types of retrieving data, via Streaming API and via Rest API. It limits the returned result number per query.

Twitter makes it possible to filter the queries. A query can be filtered on language, so we retrieve the tweets with Turkish language filter.

### 2.2 Irony

General Turkish Dictionary of Turkish Language Association<sup>4</sup> defines irony as “make mocking with a person or a situation by referring the opposite meaning of what is said”. In other words, irony transforms the polarity of an expression into its opposite. There exist various definitions of irony, however almost all of them agree in “a contrast between appearance and actual reality”.

There are three types of irony in literature namely, verbal, situational and dramatic irony.

- **Verbal irony:** This type of irony occurs when speaker says one thing but means the opposite. Verbal ironies are the intentional products of speakers. It is the most ordinary form of irony and common in daily conversations. According to literary experts, sarcasm is a form of verbal irony.

Example. I absolutely adore it when my bus is late.

I am so pleased mom woke me up with vacuuming my room this morning :)

- **Situational irony:** Situational irony is the direct result of an action that is contrary to the desired or the expected outcome. Such a form of irony is the result a discrepancy in perspective, such that what is known and expected at one moment differs with what is known later on.

---

<sup>4</sup> <http://www.tdk.gov.tr>

Example. A woman has been saving painfully to buy a golden watch. Just hours after buying the watch, her daughter arrives home with the same watch as a gift for her!

- **Dramatic irony:** Dramatic irony is used in a narrative when an event occurs whose the audience understands but the characters do not. It is most commonly used by storytellers, in plays, in the theater, and in movies.

Example. A character in a series, which seems like very good person to all of the other characters, is a father, also is a police, and shows up as a murderer at the end of the series.

Neither machine learning techniques nor the humans perform well for detecting the existence of irony. Intonation, facial expression, pause duration are the clues of irony in spoken discourse. Listeners can easily detect the existence of irony with these clues while speaking. However, determining irony is more challenging in written discourse. This study is interested in the detection of only verbal ironies (sarcasm) since the context of situations and the general knowledge of the object are not considered. As a result of these, for detecting the written verbal ironies, some linguistic clues are needed which are both specific to Turkish and common for all languages.

### 2.3 SentiStrength

SentiStrength [1] is a tool for calculating sentiment score of short social texts such as tweets. This program was originally developed for English, and it can be optimized to other languages by changing the content of input files.

The results are given in three ways namely binary, trinary and single score. The results are evaluated as positive and negative for binary, positive, negative and neutral for trinary and lastly (-4 to +4) for single score. The sentiment score is expressed as given below.

[no positive emotion] 1 | 2 | 3 | 4 | 5 [very strong positive emotion]

[no negative emotion] -1 | -2 | -3 | -4 | -5 [very strong negative emotion]

SentiStrength needs its support files to work. The information in these files is used in algorithms and they can be customized. The short explanations and the name of the files are explained in Table 1.

*Table 1 SentiStrength supporting file list*

<b>File Name</b>	<b>Explanation</b>
<b>BoosterWordList</b>	Includes the booster words and sentiment scores of each of them.
<b>EmoticonLookupTable</b>	Includes the emoticon lists and scores.
<b>EmotionLookupTable</b>	Includes the words that contain emotions and sentiment scores of them.
<b>EnglishWordList</b>	Includes English word list.
<b>IdiomLookupTable</b>	Includes idioms and sentiment scores.
<b>NegatingWordList</b>	Includes the words that give negative sentiment to sentence.
<b>QuestionWords</b>	Includes the question words.
<b>SlangLookupTable</b>	Includes slang words.

The language support is provided by customizing the input files. Turkish language support and program customization were done by Gural Vural, METU Computer Engineering Department.

## **2.4 Natural Language Processing**

Natural language processing (NLP) is a research area that analyzes and understands human language. NLP aims to gather information from words how human can understand.

Therefore, NLP can be considered a part of Artificial Intelligence. The most challenging task of NLP is deriving meaning from sentences, words as humans.

Sentence segmentation, sentiment analysis, part-of-speech tagging and information retrieval are the major tasks of natural language processing. Sentiment analysis is a subtask of NLP to extract sentimental information from the texts.

NLP analyzes the text in three different ways:

- It can analyze the words morphologically with splitting the word to roots and suffixes, simply concentrated to structure of the word.
- It can analyze the words syntactically to reveal the relations between the words grammatically.
- It can analyze the words semantically to gather the meaning of the word.

#### **2.4.1 Part of Speech**

In grammar, part of speech is simply a category of a word. In other words, it is not what the word is, it is how the word is used. Nouns, verbs, adjectives and adverbs are the part of speech categories.

Part of speech tagging is the process of labeling the words with the corresponded part of speech category.

#### **2.4.2 Zemberek**

Zemberek is an open source NLP library that designed for analyzing Turkish texts morphologically. The library has the abilities of spell checking, stemming, and correction of the mistyped words [2].

Zemberek does the word analysis, in other words it splits the word into root and suffixes. For this study, type of the suffixes is important since they can represent sentiment of word. Zemberek gives the all possible analysis of words and an example is shown for the word “bulamaz” in Figure 1.

```
Cözümlenmeler:  
[ Kok: bul, FIIL ] Ekler: FIIL_YETERSIZLIK_E + FIIL_OLUMSUZLUK_ME + FIIL_GENISZAMAN_IR  
[ Kok: bul, FIIL ] Ekler: FIIL_ISTEK_E + FIIL_OLUMSUZLUK_ME + FIIL_GENISZAMAN_IR  
[ Kok: bula, FIIL ] Ekler: FIIL_OLUMSUZLUK_ME + FIIL_GENISZAMAN_IR  
[ Kok: bula, FIIL ] Ekler: FIIL_DONUSUM_MEZ  
[ Kok: bu, ZAMIR ] Ekler: ISIM_DONUSUM_LE + FIIL_OLUMSUZLUK_ME + FIIL_GENISZAMAN_IR  
[ Kok: bu, ZAMIR ] Ekler: ISIM_DONUSUM_LE + FIIL_DONUSUM_MEZ
```

Figure 1 An example of splitting word “bulamaz”

For correction of mistyped words, Zemberek checks the input word and decides whether it is written correctly. If it is not written correctly, Zemberek suggests the list of possible words. An example is shown for the word “önrme” in Figure 2.

```
önrme için öneriler:  
önüme  
öneme  
önerme  
örme
```

Figure 2 Correction of mistyped word “önrme”

## 2.5 Machine Learning

Machine learning (ML) is interested in design and development of system that can learn and improve its performance [3]. Machine learning is a branch of artificial intelligence and the systems that are constructed with machine learning are trainable.

A system generated with machine learning needs training dataset for learning. It extracts features from data and converts them into knowledge. After that, it uses extracted knowledge for improving the performance and predicting the output of non-trained data.

Machine learning algorithms are diversified into types in terms of the type of the input data or the desired outcomes. Supervised learning, unsupervised learning, semi supervised learning, transduction, reinforcement learning and development learning are the types of the machine learning algorithms. First three of these categories are related to type of the input data and the rest are related to outcomes of the algorithms. In this study, supervised learning is used since classification is required. Supervised learning methods require labeled data for training and use the classification for new unlabeled data.

### 2.5.1 Naïve Bayes Classifier

Naïve Bayes Classifier [4], based on Bayes' theorem, is widely used in text classification. It is simple, computational efficient and has good classification performance. These are reasons of being suitable for text classification.

Theoretically, Naïve Bayes classifier claims that the presence or absence of particular feature is not related to the presence or absence of another feature. In other words, each of the features contributes independently to the probability.

Naïve Bayes classifier is suitable for high dimensional input data. Also, it requires small amount of training data for making estimation. It is the advantage of this algorithm.

This classifier is using the joint probabilities of classes in the dataset. By this way, it will be able to estimate the new one. Assume  $x = (f_1, f_2, \dots, f_n)$  is an element where  $f_i$  represents one of the features between n elements and c is a class. The probability of x to be in class c is shown in Equation 2.1.

$$p(c|(f_1, f_2, \dots, f_n)) = \frac{p(c) p((f_1, f_2, \dots, f_n)|c)}{p((f_1, f_2, \dots, f_n))} \quad (2.1)$$

The denominator does not depend on c and the values of the features  $f_i$  are given, then denominator becomes a constant. And the nominator, which shows joint probability model, can be rewritten using the chain rule. It is shown in Equation 2.2.

$$p(c|(f_1, f_2, \dots, f_n)) = p(c)p(f_1|c)p(f_2|c)p(f_3|c) \dots p(f_n|c) \quad (2.2)$$

The assumption of independence of the features is the naïve part of this algorithm. Now, assume  $f_i$  is a feature which is conditionally independent from the feature  $f_j$  where  $i \neq j$ . It is shown in Equation 2.3.

$$p(c|(f_1, f_2, \dots, f_n)) = p(c) \prod_{i=1}^n p(f_i|c) \quad (2.3)$$

## 2.5.2 Support Vector Machines

Support vector machine (SVM) is a group of supervised learning methods that can be applied to classification or regression [5]. It is based on the concept of decision planes, which are a separator between with set of objects having different class membership.

SVM is basically suitable for binary classification problem. Assume a set of training data, which is assigned to one of the two groups, is given, new input will be assigned to right class. A data point is a list of  $p$  numbers,  $p$  dimensional vector, and this algorithm tries to separate this points with  $(p-1)$  dimensional hyperplane. Best hyperplane provides the largest separation. Moreover, SVM is also used for multiclass classification. The flow of the algorithm does not change. Therefore, the classification will be binary again, the data classified as the group that is wanted and the rest of all groups.

The algorithm will be explained as follows.

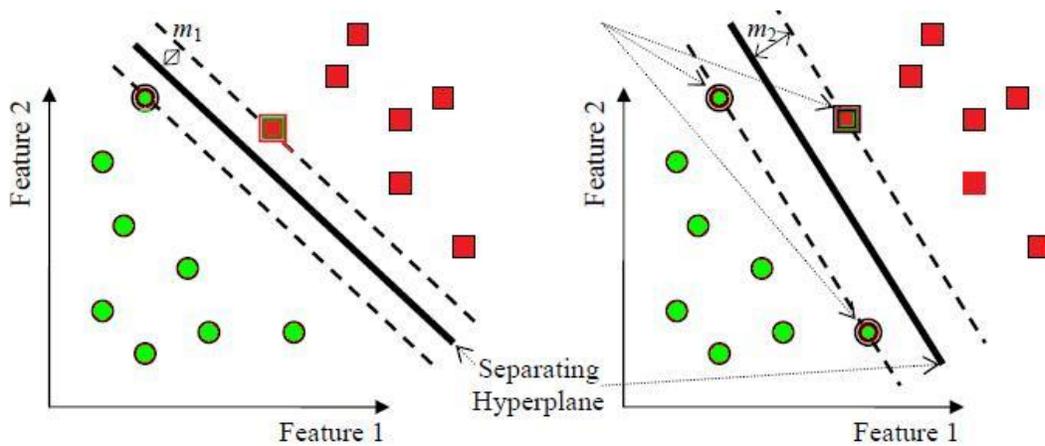


Figure 3 The change of hyperplane when classifying with SVM

If we look at above two dimensional spaces, assume that data has to map two categories which are represented by rectangles and circles. In this case, it can be clearly seen that data can be separable and circles are on the lower left corner, rectangles are upper right corner. The SVM responsible for find a line (a one-dimensional hyperplane) that separates the data by means of their labeled categories. Unfortunately there are a lot of lines separating them, then this gets the questions on mind: “Which line is better?”

In the Figure 3 two dashed lines drawn parallel to the separating line with passing through the closest vectors to showing the distance from the separating line. This distance is called margin and the vectors that are passed through are named support vectors.

An SVM analysis can find a line, which can be called generally hyperplane, which is lying along the path so that the margin between the support vectors will be maximized.

The advantages [6] of SVM can be listed as follows:

- High accuracy
- Effectiveness in high dimensional spaces even if when number of dimensions is greater than the number of samples
- Memory efficient
- Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of SVM can be listed as follows:

- Poor performance when the number of features is much greater than the number of samples
- Evaluation (five-fold cross-validation) is costly since SVMs do not directly provide probability estimates

### **2.5.3 K-Nearest Neighbors**

K-nearest neighbors (k-NN) algorithm is one of the simplest machine learning algorithms [7]. It is a non-parametric lazy learning algorithm and from the beginnings of 1970's it has been used in statistical estimation and pattern recognition.

K-NN assumes that the data used is in feature space, so they can be scalars or vectors. In addition, "k" is a user defined constant which is used for deciding how many neighbors influence the classification. It will be chosen with respect to the data and classification number. In order to choose a good "k" value, some heuristic techniques will be applied. When classification will be done between two groups (binary), then "k" is selected as 1

for best results and it is simply called as nearest neighbor algorithm. The larger values of “k” may ruin the success of the algorithm.

This algorithm has a simple logic behind it. K-NN algorithm is based on calculating the distance between a test sample and the specified training samples. Euclidian distance is used for this algorithm mostly. The training set must be in at least two groups. A sample from test set is assigned to correct group by calculating the distance between this sample and the other ones in training set.

The advantages [8] of k-NN can be listed as follows:

- Simple and intuitive
- Effective even if the training data is large
- Competitive classification performance
- Robust to noisy data

The disadvantages of k-NN can be listed as follows:

- When the training data is large, it has poor runtime performance.
- Since it is a distance based algorithm, it is hard to find which type of distance algorithm gives the best results.
- Since it is a distance based algorithm and the distance between each entry and all training data is computed, there is a non-negligible computational cost.

As a result, this algorithm gives best results when training set is used as test set since there is no training phase in k-NN.

#### **2.5.4 Random Forests**

The term “Random Forest” came from random decision forests method that was first proposed by Tin Kam Ho of Bell Labs in 1995 [9]. It was influenced from a work that proposed the idea of searching over a random subset of the available decisions and then Breiman's "bagging" idea and the random selection of features approach are combined.

The algorithm works as follows. A Random Forest consists of an arbitrary number of simple trees. These trees are used to determine the final outcome. For classification problems, the group of simple trees vote for the most popular class. Using tree ensembles can lead to significant improvement in prediction accuracy.

Random Forests are formed by growing many classification trees and the logic behind growing trees is explained below.

1. Sample  $N$  cases from the original training data. This sample will be the training set for the classification tree.
2. Select random  $m$  variables among  $M$  input variables. The selected  $m$  value must be much less than  $M$  and must be constant during the forest growing.
3. It is important to choose  $m$  optimal since it is directly related to correlation and strength of each of the trees.
4. Calculate the best split based on these  $m$  variables in the training set.
5. Grow to the largest extent possible. There is no pruning for growing trees.

The Random Forests algorithm is one of the best among classification algorithms since it is able to classify large amounts of data with accuracy. The following list shows the top benefits of this algorithm [10].

- Accuracy
- Runs efficiently on large data
- Handles large number of input variables
- Provides effective methods for estimating missing data
- Saves generated trees for future use
- Easy to set parameters



## CHAPTER 3

### RELATED WORK

In this section, the studies about irony detection are summarized. As it is stated before, there are very limited studies conducted in this area. All studies are collected and summarized in this section. Moreover, sentiment analysis is a part of this study, so the previous studies about sentiment analysis are also examined. Sentiment analysis is used for both as a feature to detect irony and as a bridge to find true testing data.

Irony detection analysis was held in Portuguese, Brazilian Portuguese, Dutch, and English ... etc. However, Turkish is not one of these languages. This study will be the first one on irony detection in Turkish.

#### 3.1 Previous Studies on Irony Detection

Carvalho et al. [11] made a study of detecting irony from user comments in online newspaper. Their data is in Portuguese language. The word irony is used for the case of an expression that has positive sentiment but expressed negative opinion. Since it is hard to detect irony in texts, they proposed eight different linguistic patterns. These patterns are explained as follows:

1. Diminutive Forms: In Portuguese, although diminutives are widely used to express affect, tenderness and intimacy, they can also be sarcastically used for expressing insult.
2. Demonstrative Determiners: Using the determiners (“this”, “that”) before human names means that person is negatively mentioned.

3. Interjections: Interjection is word used for expressing emotions and it is usually placed in the beginning of the sentence.
4. Verb Morphology: Using second-person singular as pronoun or embedded in the morphology of the verb may be a clue for irony.
5. Cross-constructions: Evaluation adjectives which have not negative polarity is a clue for having irony if they used with as cross-constructions.
6. Heavy Punctuation: Containing more than one exclamation marks or question marks in a sentence mostly is a clue for having irony.
7. Quotation Marks: Having a positive polarity sentence which is surrounded with quotation marks includes irony.
8. Laughter Expressions: Acronyms like “lol”, onomatopoeic expressions like “ah, eh” and positive emoticons “:)” are considered as having irony.

As a result, the most productive patterns are laughter expressions, heavy punctuation, quotation marks and interjections.

The study conducted by González et al. [12] interested in identifying sarcasm in tweets. They used lexical and pragmatic factors to detect the sarcasm in texts and distinguish them from positive and negative ones. The tweets that contain hashtag “#happy, #sad, #sarcasm” are collected via Twitter API and labeled as “positive, negative, sarcastic” respectively. This study also claims that sarcastic tweets mostly seem positive but have negative meaning. LIWC<sup>5</sup> Dictionary which analyzes texts as emotionally, WordNet Affect<sup>6</sup> which represents affective concepts correlated with affective words, interjections and punctuations are used as lexical factors for classification. Positive emoticons, negative emoticons and tweets that are reply to another tweet are pragmatic factors. They compare the results between human judges and lexical and pragmatic factors on machine learning techniques. Human judges state that identifying sarcasm in texts is a hard task since

---

<sup>5</sup> <http://liwc.net/>

<sup>6</sup> <http://wndomains.fbk.eu/wnaffect.html>

sometimes they need some knowledge about what is going on in the world to label the text as sarcastic. In that case, machines are fail to label that texts as sarcastic.

Tsur et al. [13] presents Semi-supervised Algorithm for Sarcasm Identification (SASI) which recognizes sarcastic sentences in product reviews in Amazon. Their dataset consists of 66000 reviews. As a training dataset, they are given 80 labeled sentences whose scores vary from 1 to 5. They try to catch helpful patterns from their training data to be having sarcasm. Number of exclamation marks, question marks, quotes and capitalized words are clues for including sarcasm. As a result, they achieved 77% on an evaluation set containing newly discovered sarcastic texts.

Another study which is conducted by Barbieri et al. [14], interested in automatic detection of irony in Twitter. They use a corpus which consists of 40000 tweets. This corpus belongs to [15]. These tweets are automatically collected in four categories via using the hashtags (#irony, #education, #humour, #politics). The hashtags are removed from the text. Their model uses seven groups of features which are listed below.

- Frequency (gap between rare and common words): Using the common and the rare English words in a tweet creates an imbalance and this may be a sign for irony.
- Written-Spoken (written-spoken style uses): A word which is used more often in written or spoken language can be a clue for irony.
- Intensity (intensity of adverbs and adjectives): The intensity values (more or less exaggerated) of adjectives and adverbs may help to detect irony.
- Structure (length, punctuation, emoticons and links): The length of the tweet, the number of words in a tweet, the number of verbs, nouns, adjectives and adverbs, the sum of the number of full stop, ellipsis, commas and exclamation marks and the number of the emoticons of a tweet may play a role for detecting irony.
- Sentiments (gap between positive and negative terms): The imbalance between the positive and the negative sentiment of a tweet may be a clue.
- Synonyms (common vs. rare synonyms use): The synonyms of each word in a tweet and their frequencies are calculated and they are used for irony detection since writers may use a term for it's another meaning.

- Ambiguity (measure of possible ambiguities): Using a word with many meanings and using a word with few meanings in a tweet creates ambiguity and it may help to detect irony.

Three datasets (Irony vs Education, Irony vs Humour, Irony vs Politics) derived from their corpus are used for experiments. 10000 ironic and 10000 not ironic data are selected for each topic. 7500 of each of them are used for training and 2500 of rest of them are used for testing. As a result, Frequency, Structure and Synonyms are important aspects for detecting irony for all of the datasets. Surprisingly, Punctuation is a good feature for detecting irony in the dataset Irony vs. Education, however it cannot distinguish ironic ones in the dataset Irony vs. Humour.

Vanin et al. [16] used patterns to detect irony in tweets. They implement 15 different patterns for each tweet that is written in Brazilian Portuguese. Some of these patterns are special for this language. They collect tweets that contain the expression “fim do mundo” (“end of the world”). It was a popular topic since 23<sup>rd</sup> December 2013 was a probable date of end of the world according to Mayan calendar. A linguist marks all the tweets as ironic or not ironic. They group similar patterns into categories, as a result they have seven categories which are listed below.

- Lists
  - List of Laughter Expression
  - List of Emoticons
- Exact Expressions
  - “só que” (only)
  - “sim,” (yes)
  - “seria” (it would be)
  - “na boa” (good)
  - “medo” | “Medo!” | “#medo” (fear)
  - “#ironia” | “#sarcasmo” | “#joking” | “#kidding”
- Part of Speech
  - ADV + ADV | ADJ + ADJ

- Part of Speech + Exact Expressions
  - “tão” + ADJ | “tão” + ADV (as)
- Part of Speech + Lists
  - ADJ + List of Emoticons
- Part of Speech + Named Entities
  - DET + ADJ + (PREP+DET) + NE
- Demonstrative Pronouns + Named Entities
  - Demonstrative Pronouns + NE
- Symbols
  - ...<EXPR>!
  - !\*|?\*|!\*?|?\*!\*

They select 370 tweets from their corpus and 269 of them are classified correctly. Laughter marks, emoticons and use of heavy punctuation are the best clues of being ironic. On the other hand, patterns related to static expressions do not perform very well for detecting irony.

Reyes et al. [15] proposed a model to detect irony. They use a corpus that consists of 40000 tweets. These tweets are automatically collected in four categories via using the hashtags (#irony, #education, #humour, #politics). This corpus is also used by [14]. The model that they proposed is based on four groups. Each group is represented by the following dimensions.

- Signatures: This feature is looking for specific textual markers and signatures.
  - Pointedness: Punctuation marks, emoticons, capital letters and quotes are the elements of this dimension.
  - Counter-factuality: This dimension is interested in implicit marks like discursive terms (Ex. about, nevertheless, nonetheless and yet).
  - Temporal compression: The elements related to opposition in time (Ex. suddenly, now, abruptly) are the main focus of this dimension.
- Unexpectedness: This feature is intended to capture incongruity and unexpectedness.

- Temporal imbalance: This dimension is for representing the degree of opposition in a text.
- Contextual imbalance: The inconsistencies in a context are the main focus of this dimension.
- Style: It is the style of the writer, in other words how the writer uses the language.
  - Character n-grams (c-grams): The affixes and the suffixes like morphological elements are captured.
  - Skip-grams (s-grams): This dimension is looking for the adjacent words and also a word sequences with a gap.
  - Polarity s-grams (ps-grams): The sequences in s-grams are a base for this dimension and it is looking for the polarity of the words like positive, negative.
- Emotional scenarios: The three dimensions which are proposed by a psychologist are trying to quantify the emotional degree of words.
  - Activation: It is the degree of being passive or active of the word.
  - Imagery: It is the degree of how easy or difficult to visualize a mental image for the word.
  - Pleasantness: It is the degree of the pleasure of the word.

Each tweet in the corpus is transformed into a feature vector. The representativeness of a given document is calculated with respect to the all dimensions for each entry. As a result, all dimensions except temporal imbalance and pointedness are the most representative features of texts, which are ironic.

Another study [17] which is interested in detecting irony in Dutch. They collected about 78000 tweets via Twitter API using only the hashtag “#sarcasme” which means sarcasm in English. They use Balanced Winnow algorithm for classification. In their test set, there exist 3.3 million tweets, only 135 of them contain sarcasm. The classifier correctly classifies 101 among 135 sarcastic tweets. These are the outcomes of the analysis.

- The tweets which contain words, not the hashtags, “sarcasme”, “ironie” and “cynisme” are resulted as containing irony.

- Positive exclamations (jippie, yes, goh, joepie, jeej, jeuj, yay, woehoe, wow) are strong clue for sarcasm.
- Intensifiers play an important role while detecting irony.  
Ex. geweldig (awesome), heerlijk (lovely), prachtig (wonderful), natuurlijk (of course), gelukkig (fortunately), zooooo (sooooo), allerleukste (most fun), fantastisch (fantastic), and heel (veeery)
- Positive words are also a strong clue for detecting sarcasm.  
Ex. fijn (nice), gezellig (cozy), leuk (fun), origineel (original), slim (smart), favoriet (favorite), nuttig (useful)

As a result, they achieve 74% success for detecting sarcasm. They also prove that the tweets that have positive polarity are most likely to have sarcasm rather than negative ones.

### **3.2 Previous Studies on Sentiment Analysis**

As it is stated above, sentiment analysis is not the main focus of this study, it is used as a supplementary information. Because, irony mostly takes place in sentence which is positive polarity but negative meaning. In addition, sentiment analysis is also used for as a feature for detecting irony.

Sentiment analysis differs from irony detection in some ways. Sentiment analysis is more popular topic than irony detection. . However, most of the studies conducted in this area are held in English.

Since the early 2000, opinion mining has become a popular research topic in natural language processing. The impact of the rapid growth of the social media (blogs, microblogs, Facebook, Twitter ... etc.) is highly related with its popularity since web became a huge database for the analysis [18].

Generally, two classed classification method was used which means labeling texts as positive and negative. Afterwards, tagging as neutral is added to this classification. As a simple approach, looking for preselected positive words in text is enough for classifying text as positive, and similarly it is applicable for negative ones. Using emoticons is helpful

and supportive for classification [19]. They preprocessed their training data in order to reduce the features. Usernames and links are replaced with special tags not to be considered as a feature. Query terms and hashtags are also processed like similar way. Repeated characters in a word are removed to 2 and this increases the possibility of the correct form of the word. Go et al. experimented Naïve Bayes, maximum entropy and support vector machines techniques. As a result, they achieved 84% accuracy with Naïve Bayes using unigram features.

Another study [20] which also gives more importance to preprocessing state is applied two level classification. In the first level, tweets are classified as polar, neutral and irrelevant. Neutral and irrelevant tweets are ignored and they are not used in the next steps. For the second level, polar tweets are classified as positive or negative. Preprocessing step is important since it extracts the relevant information from the text and surpasses the irrelevant ones. Replacing emoticons, uppercase identification, lower casing, URL extraction, detection of usernames and hashtags, identification of punctuations, removal of stop words, removal of query term, compression of words and removing skewness of in dataset are the preprocessing state steps. Similarly, the number of machine learning algorithms is also increased and they are listed like that: Naïve Bayes, Random Forest, Support Vector Machines, Sequential Mining Optimization and J48. SMO and Random forest algorithms give more than 80% accuracy for first level classification. SVM gives consistent performance for second level classification. Naïve Bayes does not perform as expected, on the other hand SVM, SMO and J48 are displayed acceptable performance.

Another study [21] which is concentrated on opinion mining in Twitter. In this study, preprocessing is applied to tweets before using classification. Firstly, they eliminate tweets that:

- Are not English.
- Have less than 5 words.
- Contain only URL.
- Have too few words apart from greeting words.

A stop word dictionary, an emoticon dictionary and an acronym dictionary are used for detecting this special words. Emoticons are detected in tweet and emoticon dictionary is used for calculating the sentiment score of found emoticons. All remaining words are transformed into lowercase. Repeated characters in the words are converted to one character. After the preprocessing state, two different methods are used for classification. First, they use Naïve Bayes Classifier to separate tweets that contain opinion from the others. The total accuracy of extracting tweets that are containing opinion is 76.8%. The other one is using unigram Naïve Bayes classifier for building the multi classifier with pre-labeled training data. The accuracy is 91%.

Working with Turkish language is challenging and very limited studies were held in Turkish. Erogul [22] has worked on sentimental analysis of Turkish texts. He used movie reviews as dataset and classified the reviews as positive, negative and neutral. In order to classify the text correctly, support vector machine algorithm is used. For stemming, Zemberek is used to split the word into roots and suffixes. The effects of n-grams, part-of-speech tagging are examined. Negation tagging and estimating threshold value are the techniques used for feature selection. As a result, this study achieves 85% accuracy on positive-negative classification.

The other study is interested in sentiment analysis on Turkish political news [23]. The effects of different machine learning algorithms are examined. Political columns are used as data source. This study differs from the former one in terms of annotated data. The dataset of the previous study contains labeled data. The movie reviews data has the scale (for example, 0-5) information, therefore the results can be compared with this information. However, news data has no feature like the other one. Similarly, Zemberek is used for stemming of Turkish words. Naïve Bayes Classification, Support Vector Machine, the N-gram based character language model and Maximum Entropy are the algorithms that are examined. The result of the study, the N-gram based character Language Model and Maximum Entropy performed better than Naïve Bayes and SVM in terms of relative performance.

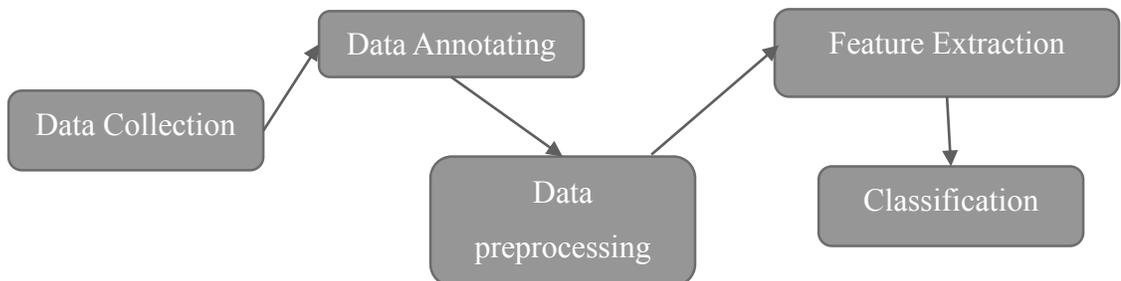


## CHAPTER 4

### METHODOLOGY

Irony detection in Turkish is the main concern of this study. In order to make classification as containing irony or not, some steps are followed. First step is data collection phase. For this phase, data is collected from Twitter via Twitter API. After that, the preprocessed data is analyzed sentimentally. Some of that data is selected for annotating. This is the time when sentiment analysis becomes a part of this study since annotated data is selected according to the sentiment scores. After this part, the next step is data preprocessing phase. All data is preprocessed towards some rules. The preprocessed data and the data which is annotated as irony constitutes the pure irony data. The next step is feature extraction phase. Features are extracted in order to detect irony in both irony data and rest of the data. The last step is classification phase. With the help of the extracted features, testing data is tried to be classified.

The flow of these steps is shown in Figure 4.



*Figure 4 The flow of methodology*

## 4.1 Data Collection

For this study, 54362 tweets are collected via Twitter API and constructed a dataset. It consists of fully Turkish tweets and they are in raw format. In other words, there is no classification information in the tweets. In order to provide the diversity in the topics of tweets, the data collection process extended over a period of time, between October 2013 and September 2014.

When collecting tweets from Twitter, no criterion is applied to them. There are a lot of tweets but only a little part of them contains irony. Sentiment analysis is performed for making candidate irony set selection wisely.

## 4.2 Data Annotating

In order to use part of the collected data as training data, it is need to be annotated. For this study, 3 people selected to annotate data. Qualtrics<sup>7</sup>, which is an open source online survey tool, is used for marking.

500 tweets are selected for training data. This data which will be annotated is constructed like that,

- 100 tweets: These are tweets which have the most positive sentiment score given by the SentiStrength.
- 100 tweets: These are tweets which have the least positive sentiment score given by the SentiStrength.
- 100 tweets: These are tweets which have the most negative sentiment score given by the SentiStrength.
- 100 tweets: These are tweets which have the least negative sentiment score given by the SentiStrength.

---

<sup>7</sup> <http://qualtrics.com/>

- 100 tweets: These are tweets which are randomly selected and have neutral sentiment score given by the SentiStrength.

There exist two categories of answers. First category has three options namely positive, negative and neutral. The people who answered the survey are told that they have to select only one of the options as an answer. The second category is for detecting irony in the text. This option has to be checked in case of thinking the text contains irony.

3 people marked tweets according to the descriptions above. All results are evaluated text by text and labeled as most selected option.

The result of the annotation is shown in Table 2.

*Table 2 The classification results of annotated data*

	<b>Positive</b>	<b>Negative</b>	<b>Neutral</b>
<b>Selected data</b>	200	200	100
<b>Annotated data</b>	219	205	76

The actual classification results are shown in the second row of Table 2 and the results generated with SentiStrength are shown in the first row. The tool SentiStrength is quite successful for classification since there is no big difference between the results. Moreover, 69 out of 500 tweets are annotated as containing irony by the judges. The distribution among these irony annotated data is shown in Figure 5. 45 out of 69 tweets are marked as positive, 10 out of 69 tweets are marked as neutral and 14 out 69 tweets are marked as negative. This situation proves that 80% of irony data has positive (and neutral) sentiment score.

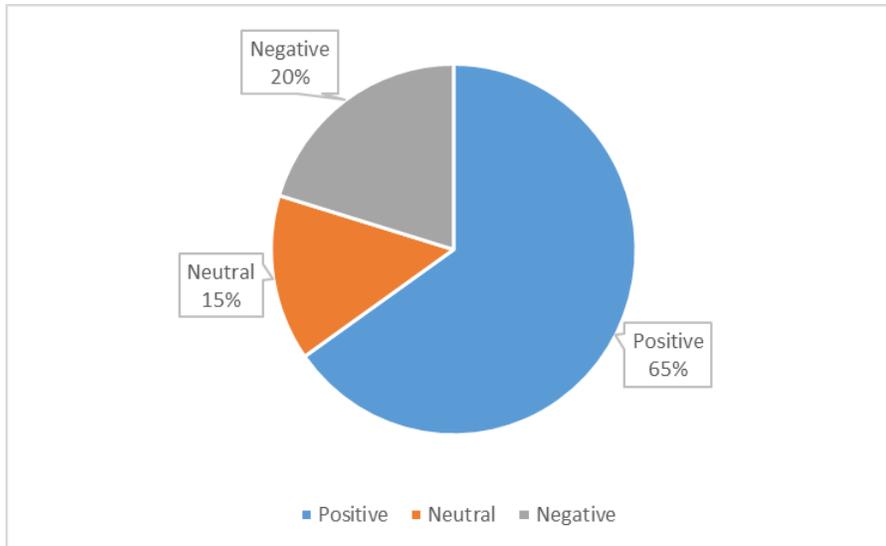


Figure 5 The sentiment distribution of irony data

### 4.3 Data Preprocessing

Tweets that are collected are preprocessed in order to gather the valuable information from them. The special keywords are assigned to some words which have no meaning for sentiment analysis to distinguish them from the others. The remaining words are applied spell checking and morphological analysis. Zemberek [2] is used as morphological analyzer.

The preprocessing steps are summarized as follows.

#### 4.3.1 URL Extraction

Some of tweets contain URLs due to the character limit of Twitter. Most of the users choose to share URLs in tweets instead of sharing their opinions about an issue. However, crawling the content of the URL and extracting sentiment from there will be too costly and is not in the scope of this study.

As a result, URLs has no sentiment information. They are replaced with the tag <URL> and these tags are not taken into consideration while classification.

### **4.3.2 Detection of Hashtags, Usernames and Retweets**

Hashtags are special words which have the character “#” as prefix. They are special keywords since hashtags group the tweets in particular subjects. Hashtags are detected from “#” character. Mostly, hashtags give no clue for sentiment analysis, so they are replaced with keyword <HASHTAG>. As an exception, some users forms the sentences with hashtags and when the hashtag is removed from the sentence, the sentence may be lose its sentiment information. These type of cases are neglected.

Usernames are special words which have the character “@” as prefix. They are the keywords that represent the usernames of users. Similarly, they are detected from “@” character. Usernames have no information for sentiment analysis and is removed from the tweets. They are replaced with the keyword <USERNAME>.

If a tweet is compelling and interesting enough, users might republish that tweet, commonly known as retweeting, and twitter employs “RT” to represent re-tweeting. Special “RT” symbol is removed from the tweets.

### **4.3.3 Uppercase Identification**

Most of the tweets are written in lowercase, it is the general habit among Twitter users. In spoken language, people can easily forward their feelings by emphasizing. However, emphasizing is not possible in written language. Therefore, people express their powerful emotions with using uppercase letters.

Also, detecting uppercase words may be a clue for detecting irony. As stated, using uppercase letters give powerful information to tweets and this information has chance to contain irony.

Unlike the first two preprocessing methods, the words containing uppercase letters are not removed from the tweets. The special keyword <UPPERCASE> is inserted before the uppercased words and they are used as a feature for detecting irony.

#### 4.3.4 Detecting Proper Names

Proper names also have no useful information for sentiment analysis. Therefore, they are removed from the tweets and replaced with the special tag <PROPERNAME>.

The method used for detecting proper names is simple. If the first letter of a word is capital letter and this word is placed in the middle of the tweet or if the word contains the character apostrophe (’), that word is most likely a proper name. As stated above, they are removed.

#### 4.3.5 Spell Checking and Correction

All of the words in tweets have to be spell checked because that words may be mistyped. Dropping vowels to shorten the word and writing the order of letters wrongly by mistake are encountered widely. In addition, the habit of not using the letters “ı,ö,ü,ç,ş,ğ” while writing causes information loss. Therefore, making spell check and doing corrections is a must.

As stated Zemberek is for these purposes. This tool has a function to check whether the word is written correctly or mistyped. Each word except the special keywords is controlled one by one. Nothing is applied to correctly written words. However, there is one more step for mistyped ones.

Zemberek also has a functionality for suggesting corrections. Mistyped words are queried for suggestions. Zemberek offers all suggestions that have Levenshtein correction distance 1 for root of the word and 1 for affix. First suggestion is considered as the most related one. As a result, mistyped word is replaced with this first suggestion. Examples of this condition are shown in Table 3 and Table 4.

Table 3 Spell checking with Zemberek (Example 1)

<b>Original sentence</b>	Gelirkn cok mutluydu.
<b>Suggestions for “Gelirkn”</b>	Gelirken, Gelirin

<b>Suggestions for “cok”</b>	Çok, sok, çök, ok, kok, şok, coş, tok, fok, cop, yok, cık, dok, lok, coğ
<b>After correction</b>	Gelirken çok mutluydu.

*Table 4 Spell checking with Zemberek (Example 2)*

<b>Original sentence</b>	Aglamaktan epey utanıyordu.
<b>Suggestions for “Aglamaktan”</b>	Allamaktan, Atlamaktan, Anlamaktan, Açlamaktan, Adlamaktan, Azlamaktan etc.
<b>Suggestions for “utanıyordu”</b>	utanıyordu
<b>After correction</b>	Allamaktan epey utanıyordu.

In the example in Table 3, the resulting sentence is become the desired form, on the other hand the meaning of sentence in Table 4 is changed after correction.

#### **4.4 Feature Extraction**

Sentiment classification simply results in three different categories, positive, negative and neutral. However, although some sentences are classified as positive, they may contain irony and principally have negative meaning. Similarly, the same situation is applicable for the sentences which are classified as negative but have positive meaning indeed. Some methods are applied for detecting these type of situations.

In order to detect irony, features that will be a clue for detecting irony are extracted. Some of these features are specific to Turkish, however some of them are general for all languages like punctuation marks. The definitions of these attributes are explained below.

#### 4.4.1 Bracketed Question Mark / Exclamation Mark

It is the most common way to express irony in a written discourse. Writers, journalists, also twitter users always use this pattern to express sarcasm in their sentences. As a result, the existence of this pattern is one of the most important clue for this study.

Ex. Evimin önüne ne kadar da güzel (!) bir çukur kazmışlar.

This sentence has positive polarity because of the adjective “güzel”. However, having this pattern reverses the meaning. The writer does not have pleasure about the activity, on the contrary, the writer feels angry about it.

Bracketed question mark is also known as “Irony mark”<sup>8</sup> which is proposed by Marcellin Jobard and French poet Alcanter de Brahm during the 19th century.

The tool searches for existence of this pattern and uses the count of it as a feature.

#### 4.4.2 Exclamation Marks

Having an exclamation mark in a sentence does not indicate being ironic. Normally, exclamation marks are used after an interjection or exclamation to indicate strong feelings. However, having repeated exclamation marks is also one of the important clues.

Ex. Evimin önüne ne kadar da güzel bir çukur kazmışlar!!!!

Similarly, this sentence has positive polarity because of the adjective “güzel” but the repeated exclamation marks changes the meaning of the sentence.

The tool searches for existence of exclamation mark and uses the count of it as a feature.

---

<sup>8</sup> [http://en.wikipedia.org/wiki/Irony\\_mark#Irony\\_mark](http://en.wikipedia.org/wiki/Irony_mark#Irony_mark)

### 4.4.3 Question Marks

Having a question mark in a sentence does not indicate being ironic. Normally, question marks are used for indicating an interrogative sentence, clause, or phrase. However, having repeated question marks is a clue for detecting irony.

Ex. Evimin önüne ne kadar da güzel bir çukur kazmışlar????

The intent of the writer of this sentence does apparently not ask a question, since having positive polarity because of the adjective “güzel”, the repeated question marks used for changing the meaning of the sentence.

The tool searches for existence of question mark and uses the count of it as a feature.

Having question and exclamation marks together (!?!?) is also a sign for sarcasm. Since the counts of these punctuation marks are calculated separately, this pattern is not used as a new attribute.

### 4.4.4 Quotation Marks

Quotation marks are used when a direct speech or title or name will be taken part in a sentence. They are also used to indicate a different meaning of a word or phrase and are often used to express irony when the word or phrase between quotation marks has a positive polarity.

Ex. Evimin önüne “ne kadar da güzel” bir çukur kazmışlar.

The quotation marks are used for injecting irony to this sentence by negating the meaning of the phrase written between them. This type of usage is known as “Scare Quotes”<sup>9</sup>.

The tool searches for existence of quotation mark and uses the count of it as a feature.

---

<sup>9</sup> [http://en.wikipedia.org/wiki/Irony\\_mark#Scare\\_quotes](http://en.wikipedia.org/wiki/Irony_mark#Scare_quotes)

#### **4.4.5 Full Stop (Ellipsis)**

Full stops, as everybody knows, are mainly used for finishing sentences. Having regular usage of full stops does not give a clue for irony. There is a special use of full stops called ellipsis (three periods of full stop). Besides its regular usage, an ellipsis indicates an unfinished thought and a nervous or awkward silence. Therefore, having repeated full stops or ellipsis sometimes used for expressing sarcasm.

Ex. Bu inanılmaz destek için gerçekten çok teşekkür ederiz!...

At the first glance, the sentence has a positive polarity. It seems like the writer feels pleasure. However, having ellipsis at the end of the sentence gives “there is nothing left to say” meaning to that sentence.

The tool searches for existence of full stops and uses the count of it as a feature.

#### **4.4.6 Smileys**

Smileys are the pictorial representation of happy faces with punctuation marks. The most common happy smileys are :), :-), (: and :)). Existence of happy smileys does not always express positive meaning. Sometimes they are used for ridiculing. List of smileys are shown in first table in APPENDIX A.

Ex. Bu hashtag de ne kadar çok tweet almış :):):)

In that sentence, the writer uses sequence of smileys to indicate irony in that sentence. The writer does not feel happy about the situation, he/she ridicules.

The tool searches for existence of a list of happy smileys and uses the count of them as a feature.

#### **4.4.7 Frowns**

Frowns, as the opposite of smileys, are the pictorial representation of sad/angry faces with punctuation marks. They are also known as sad smileys. The most common frowns

are :(, :-( and :'-(. Frowns are not an obvious clue for irony. When they are used with a sentence that has positive polarity, they become a hint for indicating irony. List of frowns are shown in second table in APPENDIX A.

Ex. Okullar açılıyor, mükemmel ya :(:(

When the emoticons in the sentence are removed, it has a positive meaning. The writer uses the frowns in order to express the irony in that sentence.

The tool searches for existence of a list of sad smileys and uses the count of them as a feature.

The use of emoticons in proper places is clues for detecting irony. However, some of the writers avoid using them. They want from the audience to understand the irony without explicit signs.

#### **4.4.8 Gap between Sentiment Scores**

Polarity of a sentence gives us very important clues about detecting irony. As it is stated before, SentiStrength is used for calculating the sentiment scores of each tweet. SentiStrength is used in binary format for this analysis so it outputs positive scores in the range of [1, 5] and negative scores in the range of [-5, -1].

Sentiment analysis is used in two aspects in this study. Firstly, the polarity of sentence is used as a supplementary information for other features. Being positive or negative is important for some features. Secondly, the gap between positive and negative sentiment scores shows an unbalance in the sentence. Using the words which have opposite polarity together is a strong clue of having irony.

Ex. Kötü oynuyor diye istenmeyen futbolcu UEFA tarafından yılın en iyi oyuncusu seçilmiş.

This sentence has 4 as positive sentiment score and -3 as negative sentiment score. The gap between them is 7 and this gap is the only clue in this sentence to be ironic. The polarity imbalance creates the irony in the sentence.

The tool calculates positive and negative sentiment scores and the difference between them are used as a feature.

#### **4.4.9 Diminutive Forms**

In general, diminutives are used for indicating the smallness of the object, expressing intimacy and complimenting. When looking in the ironical perspective, diminutives can be used for expressing an insult or depreciation towards the entity they represent. The agglutinative structure of Turkish makes the usage of diminutives possible. The diminutive suffixes are “-cik” and “-ceğiz” and they change their forms with respect to the word they append according to Turkish vowel harmony rules.

The tool Zemberek is used for detecting this feature. As it is stated before, it can provide a list of suffixes of given word. Then, each tweet is split as word by word and each word is split as the root and the suffixes. The suffixes named ISIM\_KUCULTME\_CIK and ISIM\_KUCULTME\_CEGIZ are looked in the list of suffixes of each word.

Ex. Çok acıktım çünkü bugün sadece bir kilocuk et yedim.

The word “kilocuk” contains ISIM\_KUCULTME\_CIK type of diminutive suffix. The writer insults the amount of food he/she ate however, this amount is not regarded as too little. The irony in the sentence is understood from this suffix.

The tool searches for existence of diminutive suffixes and uses the count of it as a feature.

#### **4.4.10 Interjections**

Interjections are used for conveying an emotion or a sentiment such as surprise, disgust, joy, excitement, or enthusiasm. Having interjections in a sentence is a potential clue for detecting irony. In Turkish, interjections are mostly placed at the very beginning of the sentence. The most popular interjections are “Oley”, “Bravo” and “Hey”. Most of the time, an exclamation mark is used at the end of the sentence when an interjection is used.

The list of interjections [24] used in Turkish is given in APPENDIX B.

Ex. Bravo, kapımın önüne güzel bir çukur kazmışlar!

This sentence seems like positive sentence since its positive polarity, in fact the writer is reproaching about the situation.

The tool searches for existence of a list of interjections and uses the count of them as a feature.

#### **4.4.11 Upper Case**

In Turkish, the first letter of proper nouns and the first word of a sentence are written in capitals. Generally, in Twitter people do not try to obey the grammar rules. Most of the time, tweets do not have any capital letters. They are all written in lower case. But when someone tries to emphasize a word or a phrase in a tweet, he/she writes that word or phrase in all upper case. We turn this type of misuse of grammar into an advantage and this gives us a clue about having irony.

Ex. O kadar ZEKİ ki bütün sınıfları çift dikiş gidiyor.

The word written in all capitals is emphasized. This is one the ways to express emphasize in written discourse. The irony in this sentence can be understood from this feature.

The tool searches for existence of upper case letters and uses the count of them as a feature.

#### **4.5 Classification**

After all features are extracted, it means that feature vectors are ready for classification. As it is stated before, learning and classification are done with these feature vectors. The classification algorithms and the reasons why they are selected are explained below.

- Naïve Bayes Classifier: This algorithm is chosen for since it is one of the probabilistic classifiers.
- Support Vector Machine Classifier: This algorithm is the most popular text classification algorithm and it is very successful in this work.

- K-Nearest Neighbors Classifier: This algorithm is an example of lazy learning concept. In other words, no training phase is required for this algorithm.
- Random Forest Classifier: This algorithm is chosen for since it is one of the decision trees classifiers.

Algorithms are selected in different categories in order to compare the success of them. Training data and testing data are constructed and their feature vectors are also created. Results are evaluated in two perspectives. First of all, training set is used as test set and all of these algorithms are applied and results are obtained. The other one is new testing data are introduced so all operations are done with this dataset.

## CHAPTER 5

### ANALYSIS

In this section, classification results are examined one by one. As it is stated before, Naïve Bayes Classification, Support Vector Machines Classification, K-Nearest Neighbors Classification and Random Forest Classification algorithms are used as classification algorithms in this study.

First of all, the results of classification are examined algorithm by algorithm. The effects of them and the success rates are evaluated. As a next step, the effects of attributes are examined. The importance of the features is analyzed and classification is done with subsets of attributes. Lastly, the long texts that are taken from newspaper columns are collected and same classification is performed on this long texts as a case study.

Two types of experiments are performed for each algorithm in order to show the effect of the used dataset for testing. Firstly, training dataset is used as a testing dataset since some of the algorithms produce better results for same datasets. Then, new dataset is introduced and used as testing dataset.

Balanced distribution is used for training data. There are 97 entries which contain irony and 97 entries which do not contain irony (non-irony).

As a testing set, the data is not homogenous. There are 38 irony and 59 non-irony entries in testing set. The entries for this dataset are newly introduced.

## 5.1 Evaluation Metrics

TP Rate stands for “True Positives” and shows the rate of the correctly classified instances over all instances.

FP Rate stands for “False Positives” and shows the rate of the incorrectly classified instances over all instances.

Precision stands for rate of selected items that are correct. It is calculated as the number of correct results divided by the number of all returned results. How this value is calculated is shown in Equation 5.1.

$$Precision = \frac{TP}{TP+FP} \quad (5.1)$$

Recall stands for rate of correct items that are selected. For this case, recall value is same as TP Rate value since it is calculated as the number of correct results divided by the number of results that should have been returned. How this value is calculated is shown in Equation 5.2.

$$Recall = \frac{TP}{TP+FN} \quad (5.2)$$

F-measure is a measure of a test's accuracy and the precision value (p) and the recall value (r) are considered for computing this score. It can be simply said that f-measure is weighted average of p and r. How this value is calculated is shown in Equation 5.3.

$$F - measure = \frac{2*Precision*Recall}{Precision+Recall} \quad (5.3)$$

## 5.2 Analysis on Classification Algorithms for Irony Detection

Naïve Bayes Classification, Support Vector Machines Classification, K-Nearest Neighbors Classification and Random Forest Classification algorithms are used as classification algorithms. The working principles of all of these algorithms are different

from each other. Therefore, it is intended that how the success rates changed when algorithms in different categories are used for classification.

### 5.2.1 Experimental Results under Naïve Bayes Classifier

The results of Naïve Bayes algorithm is shown in Table 5. Different test sets are evaluated separately.

Table 5 Classification results of Naïve Bayes with different datasets

	Training Set as Test Set		Test Set	
	Irony	Non-Irony	Irony	Non-Irony
Irony	79	18	25	13
Non-Irony	7	90	5	54

As it is stated before, there are 97 instances which contain irony and 97 instances which do not contain irony. When looking at the first part (training set as test set) of Table 5, 79 out of 97 instances are classified as irony and it has 81.44% success for classifying irony. 90 out of 97 instances are classified as non-irony and it has 92.78% success for detecting non-irony.

However, the results show negative change in detecting irony when new dataset is used as test set. 25 out of 38 instances are classified as irony and it has 65.78% success for this classification. This value is 81.44% for first dataset and the success rate is decreased. 54 out of 59 instances are classified as non-irony and it has 91.52% success which is pretty much the same with the other one.

Table 6 The results of Naïve Bayes Algorithm with different datasets

	Training Set as Test Set					Test Set				
	TP Rate	FP Rate	Precision	Recall	F-Measure	TP Rate	FP Rate	Precision	Recall	F-Measure
Irony	0.814	0.072	0.919	0.814	0.863	0.658	0.085	0.833	0.658	0.735
Non-Irony	0.928	0.186	0.833	0.928	0.878	0.915	0.342	0.806	0.915	0.857
Weighted Avg.	0.871	0.129	0.876	0.871	0.871	0.814	0.241	0.817	0.814	0.809

As it is seen from the Table 6, Naïve Bayes is more successful for detecting non-irony when using training set as test set. This type of success is expected already, the results are not surprising. When newly introduced entries are used as a test set, the success of this algorithm for detecting irony is decreased when compared with the other dataset.

False Positive Rate is increased when new test set is used and True Positive Rate is slightly decreased. When looking at weighted average values, the results seem like very similar to each other.

### 5.2.2 Experimental Results under Support Vector Machine Classifier

The results of Support Vector Machines algorithm is shown in Table 7. Different test sets are evaluated separately.

*Table 7 Classification results of Support Vector Machines with different datasets*

	Training Set as Test Set		Test Set	
	Irony	Non-Irony	Irony	Non-Irony
Irony	87	10	26	12
Non-Irony	4	93	2	57

Table 7 indicates that Support Vector Machine classifier is more successful than Naïve Bayes classifier. In this case, 87 out of 97 instances are classified as containing irony and 10 out of 97 instances are classified incorrectly. In other words, 89.69% of them classified correctly when considering detection of irony. The results of detecting non-irony are better than the ones of irony. 93 out of 97 instances are classified correctly and it has 95.87% success rate.

The results of irony classification of new test set resembles with the same results of Naïve Bayes algorithm. In this instance, 26 out of 38 is classified correctly and assigned to the group irony. 57 out of 59 instances are classified as non-irony and it has 96.61% success rate for detecting non-irony.

As it is seen from Table 8, true classification of irony is more successful when using training set as test set. There is a considerable difference in true positive rates between

two datasets. While the true positive rate is 0.897 for training set used as test set, it is 0.684 for new test set. True positive rates of non-irony resembles of each other.

Table 8 The results of Support Vector Machines Algorithm with different datasets

	Training Set as Test Set					Test Set				
	TP Rate	FP Rate	Precision	Recall	F-Measure	TP Rate	FP Rate	Precision	Recall	F-Measure
<b>Irony</b>	0.897	0.041	0.956	0.897	0.926	0.684	0.034	0.929	0.684	0.788
<b>Non-Irony</b>	0.959	0.103	0.903	0.959	0.93	0.966	0.316	0.826	0.966	0.891
<b>Weighted Avg.</b>	0.928	0.072	0.929	0.928	0.928	0.856	0.205	0.866	0.856	0.85

Although the false positive rates show similarity for detecting irony, they are quite different when detecting non-irony. This situation is reflected to the results shown in weighted average values. As a result, it can said that the results are more precise when training set is used as test set.

As a result, the rates of correctly classified instances which contain irony are quietly decreased when new dataset is introduced. On the other hand, the results of classification of non-irony seem very similar.

### 5.2.3 Experimental Results under k-Nearest Neighbor Classifier

The results of k-Nearest Neighbors algorithm are shown in Table 9. Different test sets are evaluated separately.

Table 9 Classification results of k-Nearest Neighbors with different datasets

	Training Set as Test Set		Test Set	
	Irony	Non-Irony	Irony	Non-Irony
<b>Irony</b>	96	1	32	6
<b>Non-Irony</b>	1	96	0	59

The results shown in Table 9 support the explanations written above. Only 1 instance is classified wrongly both irony and non-irony. 96 out of 97 instances are classified correctly, and the rate of success is 98.96% when considering the case of training set used as test set.

Table 10 The results of *k*-Nearest Neighbors Algorithm with different datasets

	Training Set as Test Set					Test Set				
	TP Rate	FP Rate	Precision	Recall	F-Measure	TP Rate	FP Rate	Precision	Recall	F-Measure
<b>Irony</b>	0.99	0.01	0.99	0.99	0.99	0.842	0	1	0.842	0.914
<b>Non-Irony</b>	0.99	0.01	0.99	0.99	0.99	1	0.158	0.908	1	0.952
<b>Weighted Avg.</b>	0.99	0.01	0.99	0.99	0.99	0.938	0.096	0.944	0.938	0.937

Table 10 shows that the results of training set used as test set are almost perfect. The true positive rates of both irony and non-irony are 0.99. As it is stated in Chapter 2.5.3 K-Nearest Neighbors, *k*-NN algorithm gives best results when using same dataset is used both training and testing. Therefore, the results of first part are not surprising. Using new test set does not reduce the success of true positive rates dramatically. For example, true positive rate of non-irony is 1, which means that all instances are correctly classified.

New test set does not influence the success of this algorithm so much. All non-irony instances are classified correctly. 32 out of 38 instances are classified as irony and the success rate of irony is 84.21%.

#### 5.2.4 Experimental Results under Random Forests Classifier

The results of Random Forests algorithm is shown in Table 11. Different test sets are evaluated separately.

Table 11 Classification results of Random Forests with different datasets

	Training Set as Test Set		Test Set	
	Irony	Non-Irony	Irony	Non-Irony
<b>Irony</b>	96	1	31	7
<b>Non-Irony</b>	1	96	2	57

The results shown in Table 11 support the explanations written above. This algorithm is also one of the most successful ones when training set is used as test set. Only 1 instance is classified wrongly both irony and non-irony. 96 out of 97 instances are classified

correctly, and the rate of success is 98.96% when considering the case of training set used as test set.

As it is seen from the second part of Table 11, 31 out of 38 irony instances are correctly classified. The success rate of this case is 81.57%. 2 non-irony instances are classified as irony and the success rate of non-irony is 96.61%.

*Table 12 The results of Random Forests Algorithm with different datasets*

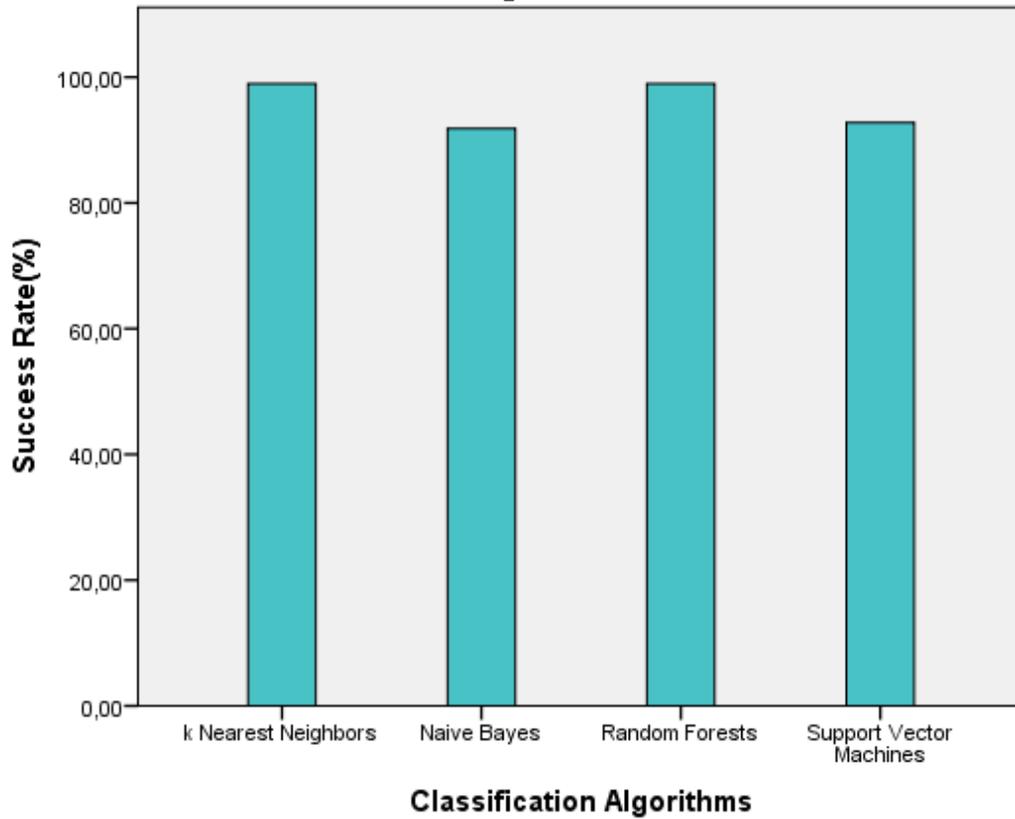
	Training Set as Test Set					Test Set				
	TP Rate	FP Rate	Precision	Recall	F-Measure	TP Rate	FP Rate	Precision	Recall	F-Measure
<b>Irony</b>	0.99	0.01	0.99	0.99	0.99	0.816	0.034	0.939	0.816	0.873
<b>Non-Irony</b>	0.99	0.01	0.99	0.99	0.99	0.966	0.184	0.891	0.966	0.927
<b>Weighted Avg.</b>	0.99	0.01	0.99	0.99	0.99	0.907	0.125	0.91	0.907	0.906

When using training set as test set, the classification is almost done without any errors. True positive rate of irony and non-irony are 0.99. Similarly, this algorithm gives best results when the same dataset is used both training and testing. The results seem like pretty successful for new dataset. The true positive rate is higher when detecting non-irony.

### **5.2.5 General View on the Performance of Classification**

Each algorithm is analyzed and explained in detail. Since the working principles of the algorithms used are different for each of them, the results also differ whether which testing set is used. Training set used as testing set scenario is shown in Figure 6.

The explanation of success rate which is used in the following figures is the number of true classified instances divided by the total number of instances.



*Figure 6 General overview of the results generated with training set is used for test set scenario*

Figure 6 shows the success rates of the classification algorithms. Since it can be seen from the figure, the most successful results are k Nearest Neighbors and Random Forests. The error rate is nearly negligible, the results of them are nearly 100%.

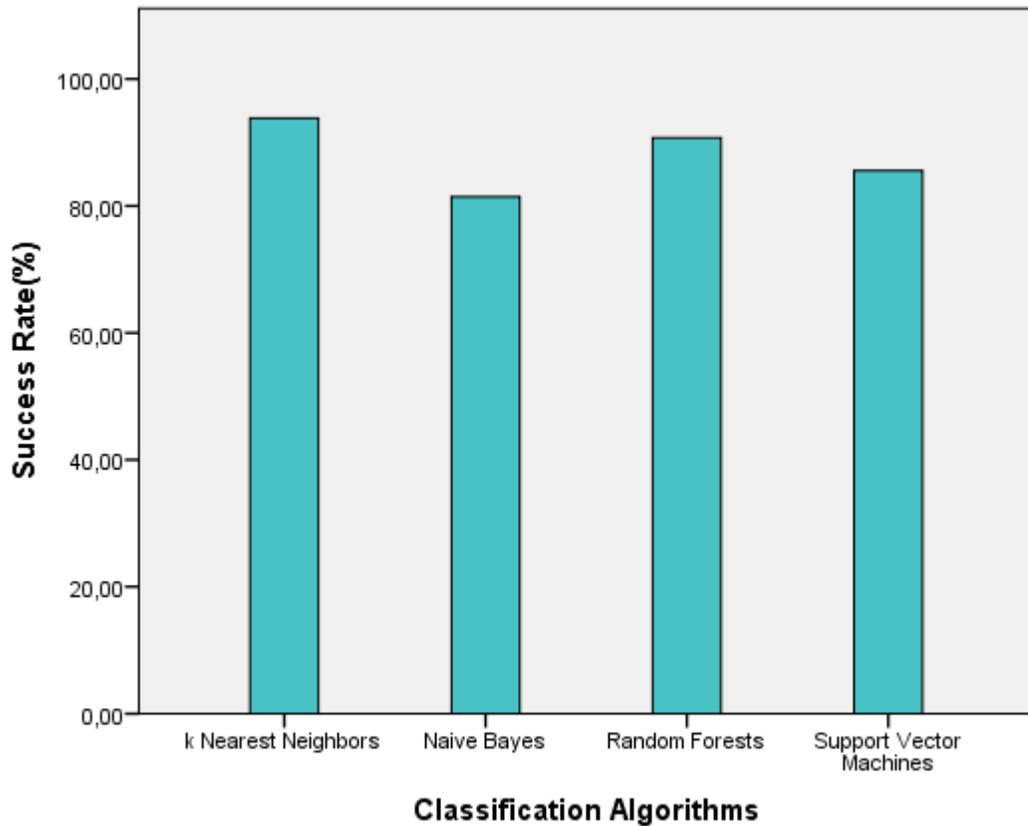


Figure 7 General overview of the results generated with newly introduced test set scenario

Figure 7 shows that k-Nearest Neighbors algorithm gives the best results when newly introduced dataset is used as test set. The results for Random Forests algorithm is almost equal to k Nearest Neighbors algorithm.

### 5.3 Analysis on the Effect of Attributes on Irony Detection

For this study, there are 11 different attributes to create feature vector. It is important to find out that which of these attributes contributes more. Two type of analysis is done for this purpose.

### 5.3.1 Experimental Results under Best First Attribute Selection Technique

For the first analysis, “CfsSubsetEval” algorithm is used as “Attribute Elevator” and “Best First” technique is used for “Search Method”. This analysis outputs top three attributes which contribute most. The results are as follows:

1. Bracketed Question Mark / Exclamation Mark
2. Exclamation Mark
3. Question Mark

The results came out as expected since the bracketed question mark or exclamation mark is the most obvious clue of containing irony. Therefore, this attribute covers the second and the third ones.

The same classification procedures are applied to same training and testing data but only these top 3 attributes are used for features. The results are shown in Table 13.

Table 13 Classification results with Top 3 attributes (Training set)

Training Set as Test Set	Naive Bayes		Support Vector Machines		k Nearest Neighbors		Random Forests	
	Irony	Non-irony	Irony	Non-irony	Irony	Non-irony	Irony	Non-irony
<b>Irony</b>	59	38	79	18	77	20	76	21
<b>Non-irony</b>	3	94	7	91	8	89	9	88

The results shown in Table 13 are obtained with the case training dataset is also used as testing dataset. The most successful results are the ones with Support Vector Machines. 170 out of 195 are classified correctly and the success rate is 87.17%.

Table 14 Classification results with Top 3 attributes (Test set)

Test Set	Naive Bayes		Support Vector Machines		k Nearest Neighbors		Random Forests	
	Irony	Non-irony	Irony	Non-irony	Irony	Non-irony	Irony	Non-irony
<b>Irony</b>	22	16	33	5	31	7	30	8
<b>Non-irony</b>	1	58	4	55	5	54	5	54

Table 14 shows the results in similar direction. The most accurate algorithm is Support Vector Machines, again. K-NN, Random Forests and Naïve Bayes follow it respectively. The success rate of SVM is 90.72%.

### 5.3.2 Experimental Results under Ranker Attribute Selection Technique

The second analysis is done with the attributes “InfoGainAttributeEval” as “Attribute Elevator” and “Ranker” as “Search Method”. This method sorts all attributes from the most important to least important. The results are like that:

1. Exclamation Mark
2. Bracketed Question Mark / Exclamation Mark
3. Gap between Sentiment Scores
4. Question Marks
5. Upper Case
6. Interjections
7. Diminutive Forms
8. Full stop (Ellipsis)
9. Frowns
10. Quotation Marks
11. Smileys

The most important 5 attributes are selected and all the analysis is performed only with these features.

*Table 15 Classification results of Top 5 attributes (Training set)*

Training Set as Test Set	Naive Bayes		Support Vector Machines		k Nearest Neighbors		Random Forests	
	Irony	Non-irony	Irony	Non-irony	Irony	Non-irony	Irony	Non-irony
<b>Irony</b>	74	23	76	21	85	12	84	13
<b>Non-irony</b>	5	92	2	95	4	93	4	93

Table 15 shows that when the attribute count is increased, the success rate of SVM is decreased. The most successful classifier is k-NN in this dataset. 85 out of 97 instances are classified correctly with k-NN and it has 91.28% totally.

Table 16 Classification results of Top 5 attributes (Test set)

Test Set	Naive Bayes		Support Vector Machines		k Nearest Neighbors		Random Forests	
	Irony	Non-irony	Irony	Non-irony	Irony	Non-irony	Irony	Non-irony
<b>Irony</b>	28	10	25	13	30	8	31	7
<b>Non-irony</b>	3	56	2	57	1	58	1	58

When newly introduced test set is used for testing, the results of k-NN & Random Forests and Support Vector Machines & Naïve Bayes are very similar to each other in tuples. But, the most accurate one is Random Forests. It has 91.75% success rate in total classification.

#### 5.4 General Overview of Effects of Attributes on Irony Detection

The analysis in section 5.2 Analysis on Classification Algorithms for Irony Detection is performed when all 11 attributes are participating. After these analysis, in order to see the effects of attribute selection, two more analysis is performed and they are explained in Section 5.3 Analysis on the Effect of Attributes on Irony Detection. All the results are

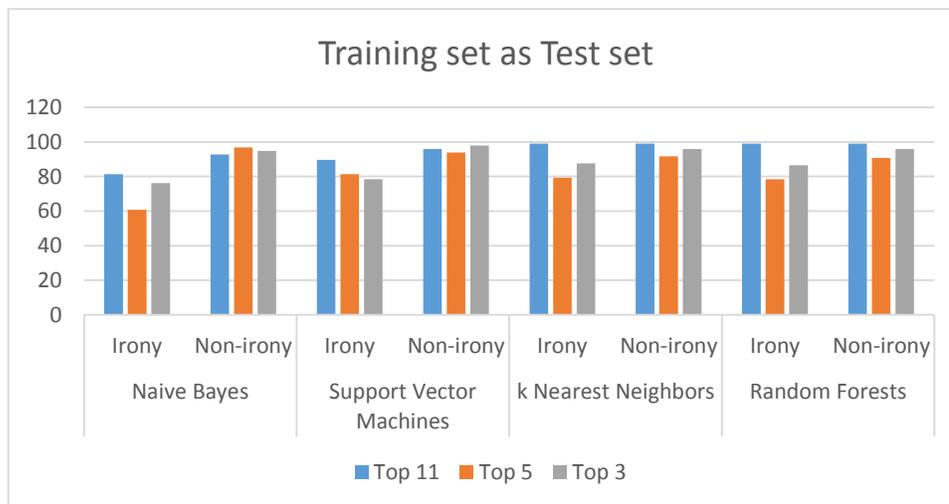


Figure 8 General overview of results of attribute selection when training set used as test set

combined and analyzed in this section.

Figure 8 shows a compact view of results of all algorithms with different attribute selection when training set is used as test set.

When classifying irony with Naïve Bayes, it is important that all the attributes are used. The most successful result is taken with when all the attributes are used. However, when classifying non-irony with Naïve Bayes, attribute selection makes no significant change in results.

When classifying irony with Support Vector Machines, the more attribute is used for classification, the more successful results are gained. The attribute count and success of results are directly related. Similar to Naïve Bayes classification, non-irony classification with Support Vector Machines is also very close to each other.

When classifying irony with k Nearest Neighbors, there is a remarkable difference between the results with all attributes and the others. In other words, the most successful results are taken with 11 attributes. The results with top 3 attributes and the results with top 5 attributes come after respectively. The ordering in the success rates is same for the classification of non-irony.

Lastly, when classifying Random Forests, the results resemble with the results of k Nearest Neighbors. The ordering of success is like the results with all attributes, the results with top 3 attributes and the results with top 5 attributes in both classifying irony and non-irony.

The explanations above are the results with training dataset are used as test dataset. Now, Figure 9 summarizes the results of all algorithms with different attribute selection with newly introduced dataset is used as test set.

Interestingly, the success of Naïve Bayes when classifying irony is highest when only top 3 attributes are used. Although the results of non-irony classification are close to each other, the result with top 5 attributes is the most successful one.

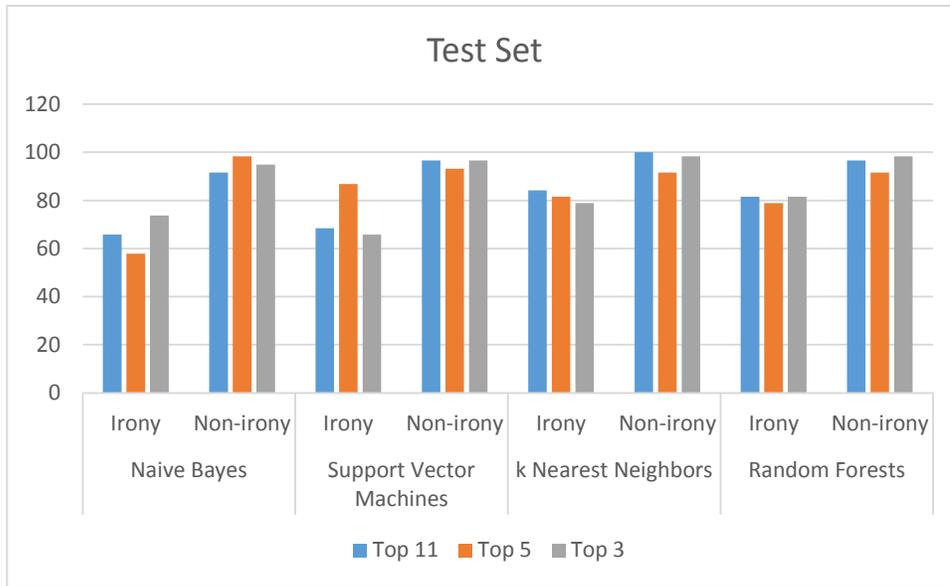


Figure 9 General overview of results of attribute selection with test set

When classifying irony with Support Vector Machines, the results of using all attributes and top 3 attributes are close to each other. However, there is significant difference between the result with top 5 attributes and the others. When classifying non-irony with Support Vector Machines, the results are so close to each other.

When classifying irony with k Nearest Neighbors, there is strong direct relationship between the success and the used attribute numbers. The results are not very different to each other, but when the all of the attributes are used, the results are the highest. When classifying non-irony, the best results are taken with all the attributes are used in classification.

When classifying irony with Random Forests, the results are so close to each other, especially the ones with all attributes and top 3 attributes are used. The same ordering is applicable for classification of non-irony, but the result of top 3 attributes used is the most successful one.

## 5.5 Effects of Model with Annotated Data

In Chapter 4.2 Data Annotating, it is stated that 500 tweets are selected to annotate. 69 out of 500 tweets are marked as containing irony. Three judges marked the tweets and the intersections of the answers of these three people are accepted as correct answer. In other words, in order to say one tweet to contain irony, at least 2 people have to say that tweet contains irony.

In order to test the success of the tool, a reverse test is applied. The training data which consists of 97 ironic and 97 non-ironic tweets is used as training data and the annotated data which consists of 69 ironic and 431 non-ironic tweets is used as test data.

Table 17 The classification results on annotated data

	<b>Irony</b>	<b>Non-irony</b>
<b>Annotated Data</b>	69	431
<b>Naive Bayes</b>	63	437
<b>Support Vector Machines</b>	73	427
<b>K Nearest Neighbors</b>	101	399
<b>Random Forest</b>	95	405

The classification results are shown in Table 17. The most consistent results are taken with Support Vector Machines. The false positive rate is higher in other algorithms except Naïve Bayes. Therefore, only the results of SVM are analyzed in detail.

SVM classifies the data as 73 irony and 427 non-irony. The difference between these result and the result of annotation is 4 tweets since remaining 69 of them are the ones that are marked as irony. These tweets are listed as follows.

- Eğer bir gün hız yüzünden ölürsem ağlamayın. Çünkü ben gülümsüyordum. -Paul Walker
- Devlet babanın dönüşü "muhteşem" oldu.... dersane ya kapanacak ya kapanacak!
- Yapmayın ya bu fairplay mi? (!)
- Telefonum suya gömüldükten sonra daha iyi çalışmaya ve daha az şarj yemeye başladı yaşasın suya düşen telefonlar!?!

These tweets are also ironic but they are not taking part in ironic tweets in annotated data. This test shows that the created model is more successful than the human judges.

## 5.6 Analysis for Irony Detection with English Dataset

In order to test the success of the model, it is decided to perform classification on different language. Therefore, English is suitable for this purpose. Since the collected training and testing data is in Turkish. English tweets are collected for this analysis. No annotation process is done. Tweets are collected in two ways. First of all English tweets are collected via Twitter Stream API and marked as not containing irony. Then, tweets that contain irony are gathered via Twitter Search API. They are queried with the hashtag “#irony” and marked as containing irony.

All features except Turkish specific ones are used for classification. As it is stated, interjections, diminutives and gap between sentiment score are the features that are specific to Turkish. SentiStrength is used for calculating sentiment scores, English version of SentiStrength is used now. Therefore only features about diminutives and interjections are not used for classification.

As a training set, 200 ironic and 200 non-ironic tweets are used. Similarly, 100 ironic and 100 non-ironic tweets are used as a testing set. The classification is performed with 9 (Exclamation Mark, Bracketed Question Mark / Exclamation Mark, Gap between Sentiment Scores, Question Marks, Upper Case, Full stop (Ellipsis), Frowns, Quotation Marks and Smileys) features and 4 algorithms (Naïve Bayes, Support Vector Machines, k-nearest Neighbors and Random Forests).

The results of training set used as a test set are shown in Table 18.

Table 18 The classification results of training data used as test data in English

Training Set as Test Set	Naïve Bayes		Support Vector Machines		k Nearest Neighbors		Random Forests	
	Irony	Non-irony	Irony	Non-irony	Irony	Non-irony	Irony	Non-irony
<b>Irony</b>	152	48	158	42	194	6	189	11
<b>Non-irony</b>	41	159	28	172	16	184	14	186

When training set is used as test set, the most successful results are observed when k-Nearest Neighbors is used for classifying. The correct classification rates of algorithms are 77.75% for Naïve Bayes, 82.5% for Support Vector Machines, 94.5% for k-Nearest Neighbors and 93.75% for Random Forests.

*Table 19 The classification results of newly introduced test data in English*

Test Set	Naive Bayes		Support Vector Machines		k Nearest Neighbors		Random Forests	
	Irony	Non-irony	Irony	Non-irony	Irony	Non-irony	Irony	Non-irony
<b>Irony</b>	66	34	61	39	68	32	69	31
<b>Non-irony</b>	17	83	18	82	18	82	24	76

Table 19 shows that the success rates of all algorithms are very similar to each other. The most successful one is k-Nearest Neighbors again and Naïve Bayes comes after. The success rates are decreased when this dataset is used.

As a result, the created model is successful for classification of irony. When different language is used, the success rates are likely to each other.

### **5.7 Case Study: Analysis on the Effect of Long Texts on Irony Detection**

Politics and sports are most popular areas to express irony in Turkey. Nowadays, there have been two elections in Turkey in last four months. First of all, the local elections were held in March 2014. After that, the presidential elections were held in August 2014. Under this circumstances, politics have been become very popular issue to talk about. While some of people were criticizing the current government, some of them showed their support.

Not only in social media but also all newspapers has become a source for expressing ideas about political situation of country. Although only the texts in Twitter are in the scope of this study, the writings of columnists can be examined as a case study. Since it is a small-scale study, the collected data is not very large. For this purpose, 5 column writings are

read and 44 sentence are selected for testing. While 27 of them contain irony, the rest 17 of them do not contain irony. The list of the selected sentences and their classifications are shown in APPENDIX C.

The training dataset that is used for previous studies is used as a training dataset again. This time, the selected sentences are used as a test set. The same methods are applied for training and testing phase.

The general view of the results is shown in Table 20.

Table 20 The classification results for case study

	Naive Bayes		Support Vector Machines		k-Nearest Neighbors		Random Forests	
	Irony	Non-irony	Irony	Non-irony	Irony	Non-irony	Irony	Non-irony
<b>Irony</b>	26	1	25	2	19	8	23	4
<b>Non-irony</b>	0	17	2	15	5	12	3	14

Naïve Bayes performs best among the other algorithms. Only one instance classified wrongly, it is assigned to non-irony class while it must belong to irony. The success rate of this algorithm is 97.72%.

The performance of Support Vector Machines is also quite well. 2 out of 27 instances are classified as non-irony while they must belong to irony. Similarly, 2 out of 17 instances are classified as irony while they must belong to non-irony. Totally, 40 instances are classified correctly and 4 instances are not classified correctly. The success rate of this algorithm is 90.90%.

K Nearest Neighbors algorithm performs worst among the other ones. The correct classification rate is 70.37% for irony and the correct classification rate is 70.58% for non-irony. Totally, 31 instances are classified correctly and the rest 13 instances are not classified correctly. As a result, the success rate of this algorithm is 70.45%.

Lastly, Random Forests algorithm shows intermediate performance for classification. 4 out of 27 instances are classified as non-irony while they must belong to irony. On the

contrary, 3 out of 17 instances are classified as irony while they must belong to non-irony. The success rate of this algorithm is 84.09%.



## CHAPTER 6

### CONCLUSION

Twitter is a good instrument to express and spread feelings in social media. Moreover expressing feelings in a 140 characters long blog, in other word a tweet, is a very hard to do thing. Most of the people say exactly what they mean in their mind. Other than that, sometimes, the others do not mean what they really wrote in their tweets. Those tweets have irony in them and this work is especially interested in finding them. Sentimental analysis stands weak to find the feeling behind the tweets with irony because most of the time ironic tweets are deceptive to sentimental analysis algorithms. The reason behind that a tweet that seems sentimentally positive with strong signs can be negative because of the nature of the irony. To decide whether a tweet contains irony, first, we research about "How can someone do irony?" we saved ironic tweets from twitter manually and we collected different ways of irony as features. This task gives us an 11 different features that will be used in algorithms that will do classification based on irony and not-irony. After that we applied four different classification algorithms, which are namely Naive Bayes, SVM, KNN, Random Forest, using saved ironic twitters as training data and 11 features. The logic behind those algorithms is different but their results are similar to each other. Numerically the least accuracy rate is Naive Bayes and it was accurate in 81%. Moreover, different analyses are performed in order to find the contribution of the attributes. Top 5 attributes are selected using the Ranker Method and Top 3 attributes are selected using Best First Method. Similarly, all the analysis is performed for these datasets. As a case study, the effects of long texts are evaluated also. The data are selected from newspaper columns. The used dataset, classification algorithms and used attributes change the results.



## REFERENCES

- [1] "SentiStrength," [Online]. Available: [sentistrength.wlv.ac.uk/](http://sentistrength.wlv.ac.uk/). [Accessed 10 01 2014].
- [2] A. A. Akın and M. D. Akın, "Zemberek, an open source NLP framework for Turkic Languages".
- [3] "Wikipedia," [Online]. Available: [http://en.wikipedia.org/wiki/Machine\\_learning](http://en.wikipedia.org/wiki/Machine_learning). [Accessed 10 01 2014].
- [4] "Wikipedia," [Online]. Available: [http://en.wikipedia.org/wiki/Naive\\_Bayes\\_classifier](http://en.wikipedia.org/wiki/Naive_Bayes_classifier). [Accessed 15 01 2014].
- [5] "Wikipedia," [Online]. Available: [http://en.wikipedia.org/wiki/Support\\_vector\\_machine](http://en.wikipedia.org/wiki/Support_vector_machine). [Accessed 13 01 2014].
- [6] "Support Vector Machines," [Online]. Available: <http://scikit-learn.org/stable/modules/svm.html>. [Accessed 20 08 2014].
- [7] "Wikipedia," [Online]. Available: [http://en.wikipedia.org/wiki/K-nearest\\_neighbors\\_algorithm](http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm). [Accessed 23 07 2014].
- [8] S. B. Imandoust and M. Bolandraftar, in *Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background*, 2013.

- [9] "Wikipedia," [Online]. Available: [http://en.wikipedia.org/wiki/Random\\_forest](http://en.wikipedia.org/wiki/Random_forest). [Accessed 29 08 2014].
- [10] "Random Forest algorithm," [Online]. Available: <http://amateurdatascientist.blogspot.com.tr/2012/01/random-forest-algorithm.html>. [Accessed 22 08 2014].
- [11] P. Carvalho, M. J. Silva, L. Sarmento and E. d. Oliveira, "Clues for Detecting Irony in User-Generated Contents: Oh...!! It's "so easy" ;-)," in *TSA*, 2009.
- [12] R. González-Ibáñez, S. Muresan and N. Wacholder, "Identifying Sarcasm in Twitter: A Closer Look," in *Human Language Technologies*, 2011.
- [13] O. Tsur, D. Davidov and A. Rappoport, "ICWSM – A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews," in *CoNLL* , 2010.
- [14] F. Barbiery and H. Saggion, "Modelling irony in Twitter: Feature Analysis and Evaluation," in *Language Resources and Evaluation Conference*, 2014.
- [15] A. Reyes, P. Rosso and T. Veale, "A multidimensional approach for detecting irony," *Language Resources and Evaluation* , vol. 47, no. 1, pp. 239-268, 2013.
- [16] A. Vanin, L. Freitas, R. Vieira and M. Bochernitsan, "Some Clues on Irony Detection in Tweets," *WWW '13 Companion*, pp. 635-636, 2013.
- [17] C. Liebrecht, F. Kunneman and A. v. d. Bosch, "The perfect solution for detecting sarcasm in tweets #not," in *Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2013.
- [18] B. Liu, *Sentiment Analysis and Opinion Mining*, 2012.

- [19] A. Go, R. Bhayani and L. Huang, "Twitter Sentiment Classification using Distant Supervision," in *LSM Proceedings of the Workshop on Languages in Social Media*, 2011.
- [20] B. Gokulakrishnan and P. Priyanthan, "Opinion Mining and Sentiment Analysis on a Twitter Data Stream," in *Advances in ICT for Emerging Regions*, 2012.
- [21] P.-W. Liang and B.-R. Dai, "Opinion Mining on Social Media Data," in *IEEE 14th International Conference on Mobile Data Management*, 2013.
- [22] U. Eroğul, Sentiment Analysis in Turkish, Msc. Thesis, METU Computer Engineering Department, 2010.
- [23] M. Kaya, G. Fidan and I. H. Toroslu, "Sentiment Analysis of Turkish Political News," in *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, 2012.
- [24] "Ünlem Listesi," [Online]. Available:  
[http://tr.wiktionary.org/wiki/Dizin:%C3%9Cnlem\\_Listesi](http://tr.wiktionary.org/wiki/Dizin:%C3%9Cnlem_Listesi). [Accessed 28 07 2014].



## APPENDIX A

### LIST OF SMILEYS & FROWNS

List of Smileys	
%-)	:-D
(-:	:-P
(:	:-}
(^ ^)	:3
(^_^)	:9
(^.^)	:d
(^_^)	:D
(o:	:P
*\o/*	:P
0:)	:X
8)	:]
:)	:b)
:))	:o)
:-)	:p
:))	:^
:-*	;)^)
:-*	<3
:-d	=)
=]	

List of Frowns	
%-(	:-\
):-:	:-
):	:/
)o:	:E
:#	:F
:('	:O
:!-(	:S
:(	:[
:((	:[
:*(	:\
:(,	:_(
:-&	:o(
:-('	:s
:-(o)	:
:-/	:•
:-S	=(
>:(	=[
>[	>/
>\	



## APPENDIX B

### LIST OF TURKISH INTERJECTIONS

List of Turkish Interjections				
a	can feda	hafazanallah	kahrolsun	ö
abe	can kurban	hah	kamera	öf
abu	cana	hahha	kardeş	öyle
acayip	cart kaba kâğıt	hain	kardeşlik	pas
adam	ce	hanımanne	keka	paydos
aferin	cık	hap	kelalaka	pehpeh
agu	cıs	hapşu	kerhaneci	pes
agucuk	cicoz	harika	keşke	pezevenk
ağababa	çabuk	haşa	keşki	pisi pisi
ağabey	çapkın	hay	kınalı kuzu	pist
ah	çer çöp	hayda	kısmet	pöf
ahbap	çinko	haydi	kış	puf
akıllı	çüş	haydin	kız	puşt
aklımda	dadaş	haydindi	köftehor	Rabbena
ala ala	dah	haydisene	köpek soyu	rahat
alabanda ateş	dalya	hayfa	kuçu kuçu	sağ ol
alarga	dandini	hayhay	kurban	sakın
aleyküm selam	dayı	hayıf	kuruyasıca	sancak alabanda
alimallah	dede	hayret	külhani	selamünaleyküm
Allah aşkına	deh	hayrola	laf	seza
Allaha ısmarladık	dehşet	hazır ol	lahavle	silah başına
Allahualem	delikanlı	hazret	lala	stop
alo	destur	hele hele	lan	süphanallah
aman	desturun	hemşehri	lava	şinanay
amanın	devam	hey	lebbeyk	tamam
amca	deyyus	heyhat	maazallah	teslim
ammena	dikkat	hiş	mafiş	teyze
âmin	dürzü	hişt	manyak	tös
ana	e	hodri	marş	tu
anam	efendi	hop	marş marş	tuh
aport	efendim	hop hop	maşallah	tuhaf

List of Turkish Interjections				
aptal	efsus	hoppala	mayna	tüh
arş	eh	hoşt	medet	tünaydın
âşık	elaman	höst	merhaba	ufaklık
aşk olsun	elveda	höt	mersi	ulan
avara	estağfurullah	hu	minnoş	üstat
ay	evlat	hurra	mirim	varda
ayı	ey	hücum	monşer	vay
ayol	eyvah	ih	mübarek	velet
babalık	fayrap	ı ih	müjde	vesselam
bacaksız	fertik	iki gözüm	müthiş	vira
bacı	fesüphanallah	ilahi	ne	viya
bakındı	fevkalade	İlahî	ne âlâ	voyvo
banko	fonda	ileri	nerede	ya
bari	fora	illallah	nereden nereye	yağlı boya
başüstüne	garip	imdat	neuzübillah	yahey
bayan	geri	insaf	neyse	yahu
be	gıdı gıdı	inşallah	nine	yaşa
bebek	gidi	isabet	ninni	yaşasın
behey	guguk guguk	iskele alabanda	o	yazık
bey kardeş	guguk kuşu	işin fenası	of	yazık günah
beybaba	güle güle	işin garibi	oğul	yetiş
bili bili	günaydın	işin kötüsü	oh	yısa
birader	güzelim	işin tuhafı	oha	yo
bonjur	ha	iyilik güzellik	olamaz	yuf
bravo	hadi	iyilik sağlık	oldu	yuh
bre	hadi hadi	kâfi	ole	yuha
can	hadisene	kâfir	oy	zıkkımın kökü

## APPENDIX C

### SENTENCES AND CLASSIFICATION RESULTS

Sentence	Class Label	Classification Success			
		Naive Bayes	SVM	kNN	Random Forest
Erdoğan gerçekten çok süper bir insan... hem ilkokul mezunu olmayı hem de profesör olmayı aynı anda yuhalatırabiliyor insanlara...	Irony	+	+	+	-
Üstelik o "devlet" dediğimiz kocaman organizma... artık Tayyibin ellerinde.	Irony	+	+	-	-
israil'e toz kondurmayan amerika, insanlık namına IŞİD'i vurmaya başlamış. insanlıkmış... peh!	Irony	+	+	+	+
sivas'ta otel yakıp katliam yapan zalimler için yaşasın cehennem! bundan böyle "kuşlar uçar, vapurlar geçer" denmeyecek...	Irony	+	+	-	+
"kuşlar uçar, minibüsler geçer" denecek.	Irony	+	+	+	+
151 işçi can vermiş, "bakan"ın derdi sosyal medyadaki eleştiriler...	Irony	+	+	+	-
kimse "bu sesle çıkma" diyememiş mi? :)	Irony	-	-	+	-
Madem senden başkasının kazanabileceğine inanmıyorsun, o yüzden mi sürekli aşağılamalar, hakaretler...	Irony	+	+	-	+
Devlet bütün gücüyle "Milletin Adamı"nı seçtirmek için uğraşıyor!	Irony	+	+	+	+
Bunlar demeyiniz mesela, ülkemizde yeterince "bunlar" diyen var, siz de ortak olmayın...	Irony	+	+	+	+
Tayyip Erdoğan'ın elinde kendisini her daim haklı ve mazlum pozisyonda tutan öyle "mükemmel bir malzeme" vardı ki...	Irony	+	+	+	+
Bravo sana, gözü "promter" cihazında olmasına rağmen İstiklal marşı yanlış okunabiliyormuş!	Irony	+	+	-	+
O değil de, Fatih Sultan Mehmet çıksa ona rakip olsa, onun bile vay haline!!!	Irony	+	+	+	+
Son iki akşamdan birinde Suriyeli mültecilerle iftar yapın. Alın onlardan birkaçını... yemezler sizi merak etmeyin.	Irony	+	+	+	+
Çünkü manzara ortada... Oy verirsin sonra yarın senin hakkında ne derler bilemem!!	Irony	+	+	+	+
Heykeltraş ağlayarak, tek tek değiştirmek zorunda kalır Türk tüccardan aldığı garantili(!) künyeleri.	Irony	+	+	+	+
Çatı adayı Mecburettin Bey'e destek verenler çığ(!) gibi büyüyor.	Irony	+	+	+	+
Mitingde "başörtülü bacım" bayıldı, asrın lideri bir dokundu şak diye ayıldı!!! mucize yani...	Irony	+	+	+	+
Bu haberi görmezden gelen şerefli(!) Türk basını, kına gecesini haber yaptı.	Irony	+	+	+	+
Yüksek hızlı trenin ismi ne biliyor musunuz? Piri Reis... İlk kez trene denizci ismi verildi!	Irony	+	+	-	+
Ben TÜRK'üm TÜRK...!	Irony	+	-	-	+

Sentence	Class Label	Classification Success			
		Naive Bayes	SVM	kNN	Random Forest
Özgürlük şarkıları söyleyen "hümanist" şarkıcımız, hapse tükilmami talep etti!	Irony	+	+	+	+
İstanbul Kongre ve gösteri merkezindeki mütevazı(!) düğüne 3000 devletli katıldı.	Irony	+	+	+	+
İyi ki Topkapı'yı gezmedi, Kaşıkçı elmasını da isteyebilirdi denildi...	Irony	+	+	+	+
Helali hoş olsun, bu mantıklı(!) izahat benim içime sinmişti.	Irony	+	+	+	+
Niye hala çekmek için senaryo aralar bu memlekette, anlamak mümkün değil hakikaten...	Irony	+	+	-	+
Reyhanlıyı, Somayı, Geziyi, ayakkabi kutularını umarım unutmamışızdır...!	Irony	+	+	-	+
Erdoğan'ın partisini ve geleceğini dizayn etme çabasının önünde hiçbir engel kalmamış olurdu.	Non-irony	+	+	-	+
Ahmet Davutoğlu'nun başbakanlığı daha da garantili olurdu.	Non-irony	+	+	+	+
Seçmenin yüzde 26'sı sandığa gitmedi.	Non-irony	+	+	+	+
Fakat buna rağmen yine de AK Parti'nin reklamcısı Erol Olçak'ın başarısını inkâr edemeyiz.	Non-irony	+	+	+	+
AK Parti'nin yanında ya da karşısında olmakla bir ilgisi yoktur. Bu başarıyı takdir etmenin hakkaniyetli olmakla bir ilgisi vardır.	Non-irony	+	-	-	-
Program sırasında dalkavuğun dalkavukluk yaptığı şahıs, kabul edilemeyecek ve tepki çekecek sözler söylüyor.	Non-irony	+	+	+	+
Dalkavuk kullanmak iyidir, hoştur, insanın kendisini iyi hissetmesine yol açar. Ama işte böyle zararlı tarafları da vardır.	Non-irony	+	-	-	-
Pardon ama Ermeni olmak, Gürcü olmaktan neden daha çirkin bir şey?	Non-irony	+	+	+	+
Başbakan'ın karşısında gerçek bir gazeteci yerine ürkekçe yaltaklanma dışında hiçbir varlık gösteremeyen bir taklacı vardı.	Non-irony	+	+	+	+
Yaradılanı gerçekten de Yaradan'dan ötürü sevip sevmediğinin kurcalanmasını, sorgulanmasını, tam olarak anlaşılmasını istemiyordur.	Non-irony	+	+	+	+
Belirsizlik işine geliyordur. Belirsizliğin daha fazla oy getireceğini düşünüyor.	Non-irony	+	+	+	+
Din, diyanet, muhafazakârlık, yaşam tarzı, başörtüsü, inanç üzerinden Ekmeleddin İhsanoğlu'na söylenebilecek tek bir kelime bile yok.	Non-irony	+	+	-	+
Artık Erdoğan'a her daim çelme takmaya çalışanların değil Tayyip Erdoğan'ın avuçlarının içinde.	Non-irony	+	+	+	+
Kime öfkelenecek, neden haykıracak, nasıl bağırarak, neyi temel alarak ortalığı inletecek?	Non-irony	+	+	+	-
Hiç kimse durup dururken vatanını, toprağını, evini terk etmez. Hiç kimse zevk olsun diye gurbet ellerde önüne gelene avuç açmaz.	Non-irony	+	+	+	+
Aldığımız tüm tepkilere rağmen "Eyvah, başımıza iş açtık galiba" demeden Filistinli mazlumların yanında dimdik durmaya devam ettiniz ya.	Non-irony	+	+	+	+
Bazen çok iyi bildiğimiz bir şeyi unuturuz, bazen bir cümleyi yanlış kurarız, bazen hata yaparız, bazen yaptığımız hatanın farkına bile varmayız.	Non-irony	+	+	-	+