ALİ BUĞRA KANBUROĞLU

M.S. Thesis

2018

GRAPH CLUSTERING APPROACH TO SENTIMENT ANALYSIS

ALİ BUĞRA KANBUROĞLU

IŞIK UNIVERSITY
2018

GRAPH CLUSTERING APPROACH TO SENTIMENT ANALYSIS

ALİ BUĞRA KANBUROĞLU
B.S., Mathematics, IŞIK UNIVERSITY, 2015
Minor., Computer Engineering, IŞIK UNIVERSITY, 2015

Submitted to the Graduate School of Science and Engineering
in partial fulfillment of the requirements for the degree of
Master of Science
in
Computer Engineering

IŞIK UNIVERSITY
2018

IŞIK UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

GRAPH CLUSTERING APPROACH TO SENTIMENT ANALYSIS

ALİ BUĞRA KANBUROĞLU

APPROVED BY:

Prof. Dr. Ercan SOLAK Işık University ________________
(Thesis Supervisor)

Assoc. Prof. Olcay Taner YILDIZ Işık University ________________

Assoc. Prof. Arzucan ÖZGÜR Boğaziçi University ________________

APPROVAL DATE: ..../..../....

GRAPH CLUSTERING APPROACH TO SENTIMENT ANALYSIS

Abstract

In this thesis, we aim at automatically predicting Turkish movie review scores using adjective clustering. We also measured the reliability of the two popular sentiment lexicons. In order to measure the agreement between these sentiment lexicons and human judgments, we designed a ranking experiment using pairwise comparisons. Then, we compared these sentiment lexicons and human judgments, and we gave results that show a moderate level of agreement between lexicons and human judgments. Furthermore, we performed adjective clustering task and singleton scoring to automatically assign scores to Turkish movie reviews. Adjective clustering reached an accuracy of 76%, singleton scoring reached an accuracy of 79%.

Keywords: Pairwise comparison, human judgment, adjective clustering

DUYGU ANALİZİNE ÇİZGE KÜMELEME YAKLAŞIMI

Özet

Bu tez çalışmasında, Türkçe film yorumlarının puanlarını sıfat kümelemesi kullanarak otomatik olarak tahmin etmeyi amaçladık. Ayrıca, popüler iki duygu sözlüğünün güvenilirliklerini ölçtük. Bu duygu sözlükleri ve insan tahminleri arasındaki uyuşmayı ölçmek için, ikili karşılaştırmalar kullanarak bir sıralama deneyi tasarladık. Ardından, bu düşünce sözlükleri ve insan tahminleri arasında karşılaştırma yaptık ve sözlükler ile insan tahminleri arasında orta seviyede bir uyuşma olduğunu gösteren sonuçları verdik. Üstelik Türkçe film yorumlarına otomatik olarak puan atamak için, sıfat kümeleme ve tekil puanlama çalışmalarını yaptık. Sıfat kümelemesi 76%'lık bir doğruluk oranına ulaşırken, tekil puanlama 79%'luk bir doğruluk düzeyine ulaştı.

Anahtar kelimeler: İkili karşılaştırma, insan tahmini, sıfat kümeleme

Acknowledgements

First of all, I would like to thank my thesis advisor Prof. Dr. Ercan Solak for being a guide and supporting me in my research studies. I would also like to thank Assoc. Prof. Banu Uzun who is the head of the Department of Mathematics for never giving up her support.

I would like to thank Hakan Kapucu, the head of library department of Işık University, Servet Cankurt the library manager and Caner Kubat who is an expert librarian for their support.

Finally, I express my special thanks to my father Özer and my mother Emine who supported me at every stage of my life, my sister Mısra whom I grew up with like a friend, and my dear fiancee Büşra who will share rest of my life.

To My Family…

Table of Contents

List of Tables

List of Figures

List of Abbreviations

NLP	**N**atural **L**anguage **P**rocessing
POS	**P**art **O**f **S**peech
GUI	**G**raphical **U**ser **I**nterface
JSON	**J**ava **S**cript **O**bject **N**otation

Chapter 1

Introduction

The widespread use of social media and the acceleration of communication channels are highly affecting society. While people are already well informed by hourly news, they are now able to receive instant news. Even if news is broadcast on a live television or radio broadcast, it takes time to post new news. Therefore, the internet became the fastest communication channel. Through social media channels such as Twitter, instant messages can be written by people on events, situations and various topics. This situation is more important for companies.

Companies are developing sales strategies for themselves by considering positive or negative comments written in social media. With respect to these positive or negative comments, companies evaluate these comments as feedbacks. People's feelings and thoughts about products, services, companies etc. can be predicted by automatic sentiment analysis.

Sentiment analysis, also called opinion mining, is the field of study that analyzes people's opinions, sentiments, appraisals, attitudes and emotions through opinions expressed in written texts. It is also subfield of Natural Language Processing (NLP). The opinion topics can be services, products, events, issues or individuals. Sentiment analysis has many different tasks such as; sentiment analysis, opinion mining, opinion analysis, opinion extraction, sentiment mining, subjectivity analysis, emotion analysis and review mining. "Sentiment analysis" term first appeared in [1].

In this thesis, we first conducted an experiment to measure the agreement between the polarities of two popular sentiment lexicons and the representations of human judgments. Secondly, we performed adjective clustering in order to assign same polarity to adjectives that are in same cluster. We applied the clustering to automatically infer review scores from given textual movie reviews.

1.1 Organization

This thesis is organized as follows:

In chapter 2, we give previous works in literature related to our present work which is experimental evaluation of two sentiment lexicons and movie review analysis.

In chapter 3, we describe an experiment on two popular sentiment lexicons. We give results and evaluations comparing sentiment lexicons to human judgments.

In chapter 4, we propose a graph clustering method in order to predict the scores of movie reviews.

In chapter 5, we discuss our methods and their results.

In chapter 6, we conclude the chapter giving all results and evaluations that are mentioned in chapter 3 and chapter 4. Then, we explore what we can do in future works.

Chapter 2

Related Works

In the study of Dalitz et al. [2], they describe how paired comparison method can be used for different purposes such as; building a sentiment lexicon and adding new words to the lexicon. In their paper, 10 different test subjects conducted two different experiments. The first one is direct assignment. In direct assignment, there is a GUI for score assignment with five degree scale for each word. The second one is paired comparison that consists of two-fold paired comparisons also presented in a GUI. Their results show that paired comparison is useful for comparing.

Godbole et al. [3] present a system which assigns scores indicating positive and negative opinion to each entity in the text corpus. Their system consists of two phases. First phase is sentiment identification and the second phase is sentiment scoring. As a result, they evaluated the importance of their scoring techniques over both news and blog entities.

Okanohara et al. [4] propose a novel type of document classification task. They conducted experiments with book reviews that have five-point scale rating. Their task is based on machine learning. They determined the scores of reviews and called the scores “sentiment polarity score” (sp-score). Then, they compared their experimental results with human performance.

Chapter 3

Experimental Evaluation of Two Sentiment Lexicons

In this chapter, we conduct an experiment that assesses the validity of prior polarities and evaluates the prior polarities of sentiment lexicons against human judgment. To begin with, we explain the two popular sentiment lexicons that we use in our experiment. Then, we state our problem and give our methodology that consists of pairwise comparison using binary insertion sort to enforce consistency. Furthermore, we design our experiment with a ranking task through pairwise comparisons and the sentiment lexicons. As a conclusion, we evaluate experimental results using Spearman correlation coefficients.

3.1 Sentiment Lexicons

For sentiment analysis, words and phrases that convey positive or negative sentiment are instrumental. Positive sentiment words such as “beautiful”, “wonderful” are used to express positive states, negative sentiment words such as “bad”, “poor” are used to express negative states [1]. As a collection of these words, there are sentiment lexicons. In many sentiment analysis problems, polarities supplied by sentiment lexicons are used. For our studies, we used two popular sentiment lexicons which are SentiWordNet [5] and SenticNet [6].

3.1.1 SentiWordNet

SentiWordNet is a lexical resource to support the sentiment classification and opinion mining applications [5]. It describes the words which are from WordNet [7] with three labels Objective, Positive and Negative for opinion mining. In other words, it assigns to each synset of WordNet numerical scores. Each scores of labels are in range from 0.0 to 1.0. SentiWordNet is freely available to use it for research studies on `http://sentiwordnet.isti.cnr.it/`.

3.1.2 SenticNet

SenticNet is a publicly available resource for sentiment analysis and opinion mining. It is inspired by SentiWordNet, but it is different than it to give polarity [6]. It uses common sense concepts. While each synset has three values in SentiWordNet, each concept c is associated to just one value $p_c \in [-1, 1]$ representing its polarity in SenticNet. In order to use SenticNet in real world applications, it is available on `http://sentic.net/`.

3.2 Problem Statement

In sentiment analysis, it is important to make sure the polarities are close represention of human judgments. Assigning negative, neutral and positive polarities by human subjects is easier rather than assigning numerical polarities. For example; “great” is positive adjective in terms of its polarity, but its numerical polarity is 0.857 by SenticNet. Our purpose is to assess the validity of the SenticNet and SentiWordNet sentiment lexicons for their use in sentiment analysis studies.

3.3 Methodology

In our methodology, we chose a ranking task in order to isolate the affects of prior polarities from other context aware algorithms. It is difficult and time-consuming to sort a list of words in terms of their polarities for humans. Therefore, we use pairwise comparisons for sorting. In addition, we use binary insertion sort [8] to reduce number of average comparisons.

3.3.1 Binary Insertion Sorting

In pairwise comparison, a person compares $n(n-1)/2$ pairs in a list of n words. However, if we use binary insertion sort, the average comparisons will be reduced to $n \log n$. For the implementation of binary insertion sort that consists of two parts which are BinarySearch and InsertionSort, we use the Algorithm 1.

Algorithm 1 Binary insertion sort algorithm

```
function BinarySearch($arr, item, low, high$)
    if high $\leq$ low then
        if item $>$ arr[low] then
            return $low + 1$
        else
            return $low$
        end if
    end if
    $mid \leftarrow (low + high)/2$
    if item == arr[mid] then
        return $mid + 1$
    end if
    if item $>$ arr[mid] then
        return $BinarySearch(arr, item, mid + 1, high)$
    end if
    return $BinarySearch(arr, item, low, mid - 1)$
end function
function InsertionSort($arr$)
    for $i = 1$ to $arr.length - 1$ do
        $j \leftarrow i - 1$
        $selected \leftarrow arr[i]$
        $location \leftarrow BinarySearch(arr, selected, 0, i - 1)$
        while j $\geq$ location do
            $arr[i] \leftarrow arr[i - 1]$
            $j \leftarrow j - 1$
        end while
        $arr[i] \leftarrow selected$
    end for
end function
```

3.4 Experiment Design

In our experiment, we used highest frequency adjectives from Brown corpus [9]. By using SenticNet and SentiWordNet resources, we found their polarities and eliminated the adjectives that do not have a polarity. In order to obtain a uniform distribution of adjectives in the polarity range of [-1, 1], we selected the adjectives which have highest and lowest polarities. Then, we assigned them as practical limits of polarity range. We divided this range to equal 19 intervals, and we chose 18 more adjectives with their polarities which are close to middle of each interval. So that, we had 20 adjectives that are equally distributed in [-1, 1] with their polarities.

Firstly, we used SenticNet polarities to divide the interval. We started to select highest and lowest polarities which are 0.945 for “incomplete” adjective and -0.940 for “unconscious” adjective from Brown corpus. Then, we calculated the interval using equation 3.1.

$$I = \frac{b-a}{n-1} \tag{3.1}$$

where a is the highest polarity which is 0.945 for this experiment, b is the lowest polarity which is -0.940 for this experiment, n is the number of selected adjectives, $n = 20$ for our experiment, and I is the interval. For these values, when we calculated I, $I = 0.099$ for each interval as shown in Figure 3.1.

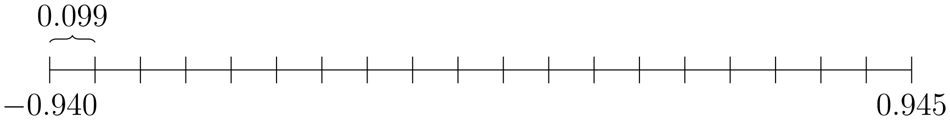

Figure 3.1: Divided intervals with highest and lowest polarities

We have total 20 adjectives, for each adjective we stated a polarity. For each polarity of adjective, we denoted by P_i. At the beginning, $P_1 = -0.940$ and $P_{20} = 0.945$. We calculated for others using equation 3.2.

$$P_i = P_{i-1} + I \tag{3.2}$$

where P_i is the polarity of i^{th} point, i takes numbers from 2 to 19, and I is interval.

Around of each P_i with ±0.05 standard deviation, we found some adjectives in terms of polarities. We selected appropriate ones among these.

The adjectives in terms of polarities both SenticNet and SentiWordNet that we use in our experiment are given in Table 3.1.

Adjective	**SenticNet**	**SentiWordNet**
romantic	0.928	0.068
great	0.857	0.259
appropriate	0.750	0.000
new	0.653	0.127
important	0.547	0.499
close	0.455	0.111
terrific	0.357	0.090
vital	0.252	0.260
real	0.154	-0.019
social	0.055	-0.008
late	-0.030	-0.025
unusual	-0.150	-0.181
difficult	-0.240	-0.708
bad	-0.360	-0.570
urgent	-0.430	0.000
white	-0.540	0.027
hot	-0.630	0.001
dry	-0.740	-0.212
permanent	-0.840	-0.625
horrible	-0.930	-0.625

Table 3.1: Prior adjective polarities.

3.4.1 Implementation of Web Interface for Pairwise Comparison

In order to collect human judgments of polarities, we have implemented a web interface which is based on binary insertion sort for pairwise comparisons. This web interface is implemented in JavaScript using binary insertion sort algorithm.

When people enter on this web interface, they firstly see the page as shown in Figure 3.2.

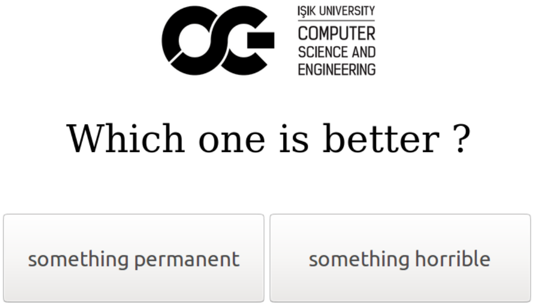
Which one is better ?

Figure 3.2: Experiment GUI for pairwise comparison

In this interface, we ask them "Which one is better ?" question because "better" adjective is neutral for judgment preference. Also, we put adjectives on screen in neutral NP constructions such as "something ..." to focus the attention of subjects on an aspect of adjectives rather than nouns. For this pairwise comparison, average experiment time is 5 minutes, and average pairwise is 40 for each subject.

3.4.2 Human Judgments of Polarities

After the design and implementation of web interface, we prepared a list of subjects who have their emails on the internet publicly. This list contains 400 native English speakers who work in different universities in Turkey. Then, we sent out emails to these native speakers to invite them to participate in the experiment.

In our email, we explained the details and purpose of our experiment, and we shared our experiment web interface url with a unique key for each subject. When they complete the pairwise comparison, their results stored in a database associated with web interface as illustrated in Table 3.2. As a result, we had responses from 43 subjects.

Key	**Ranking**
30maF3WtGmOpa6yu6QmZ	horrible, bad, late, urgent, dry, permanent, white, hot, difficult, unusual, social, real, close, new, important, romantic, appropriate, great, vital, terrific
bJiw9FvIclWjh0bMl5f2	horrible, bad, difficult, urgent, late, hot, close, vital, unusual, dry, white, permanent, social, real, new, terrific, important, appropriate, great, romantic
fUCCd902xkVFePerkDfI	late, vital, social, close, dry, urgent, horrible, permanent, difficult, appropriate, terrific, real, white, romantic, great, hot, bad, important, new, unusual
qfcEX1jYbHiwwh58OQM3	horrible, dry, bad, urgent, hot, difficult, unusual, terrific, close, romantic, late, great, real, social, white, new, important, vital, permanent, appropriate
wzrLPBtxIC0Kwzo9U7UK	horrible, bad, terrific, urgent, dry, hot, white, close, permanent, late, difficult, unusual, social, new, romantic, real, great, important, appropriate, vital

Table 3.2: Human rankings with their unique keys

3.5 Results and Evaluation

The ranking results that we collected from 43 native speakers through pairwise comparisons and the binary insertion sort are in categorical form. Figure 3.3 shows the distribution of human rankings in three dimensions.

In order to show the variations in the responses, we converted the categorical form of ranks to numerical values. For this convertion, we performed numerical assignment as below.

We assumed that adjectives and their ranks for i$^{\text{th}}$ subject as (a_j, r_j^i), where $1 \leq r_j^i \leq 20$ and a_1 is "romantic" and a_{20} is "horrible". Assuming the lowest

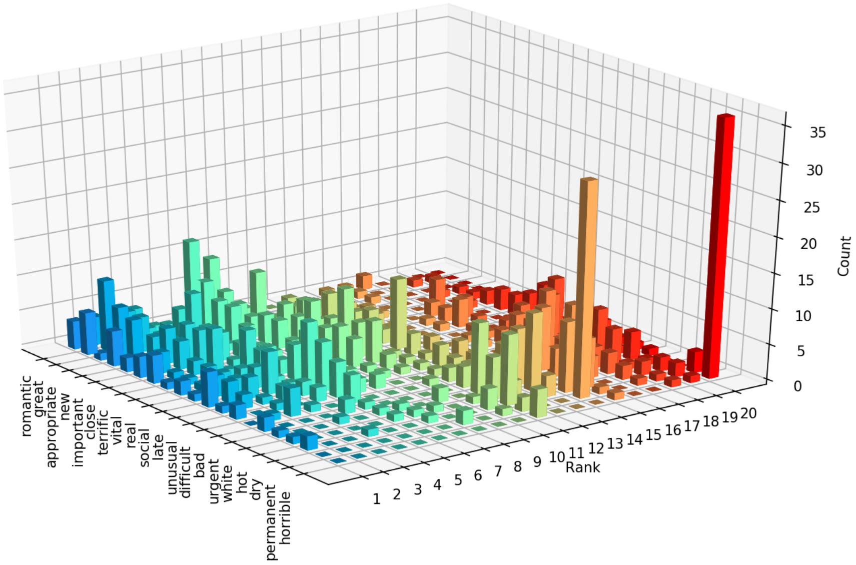

Figure 3.3: Distribution of human rankings.

ranked adjective is assigned the lowest polarity -1 and similarly, the highest ranked a polarity 1, we linearly assign each rank a polarity p^i_j within the range $[-1, 1]$ as

$$p^i_j = \frac{21 - 2r^i_j}{19}.$$

The distribution of mean polarities assigned by the responses of the 43 subjects and the standard deviations around the means are shown in Figure 3.4.

When we look at the variation in human polarity judgments, there is important amount of variation in the polarities assigned by subjects. However, 36 subjects selected the "horrible" as the lowest rank. Therefore, there is no important variation for it. For other assigned polarities, they seem more moderate. For positive polarities, they cluster around 0.5, for negative polarities they cluster around -0.5 which means a disparity to polarities obtained from SenticNet.

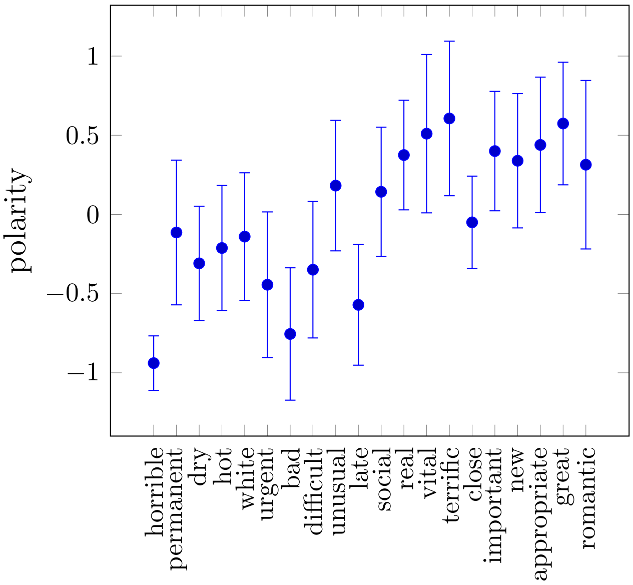

Figure 3.4: Variation in human polarity judgments

3.5.1 Spearman Correlation Coefficient

We used Spearman correlation coefficients [10] between the SenticNet and SentiWordNet rankings and human rankings in order to quantify the disparity. For two ranking experiments with n observations in each, Spearman correlation coefficient ρ is given as equation 3.3.

$$\rho = 1 - \frac{6\sum d_i^2}{n^3 - n} \tag{3.3}$$

where d_i is the difference of ranks of an observation in two experiments. The closer ρ is to 1, the more aligned are the two rankings.

First of all, we compared the polarity rankings between SenticNet and SentiWordNet using Spearman correlation coefficient, then we obtained a ρ value of 0.71 that shows a high correlation between these two lexicons.

Moreover, we compared human rankings with SenticNet. Average value of ρ across 43 subjects is 0.56 and with a confidence level of 0.01 and 18 degrees of freedom, the critical value for ρ is 0.56. Hence, we can almost reject the null hypothesis that SenticNet and human polarity judgments are not correlated.

Spearman coefficients for this comparison are shown in Figure 3.5. Furthermore, the same comparison done between human rankings and SentiWordNet with an average value of ρ is 0.53. In Figure 3.6, prior polarities are given for this comparison.

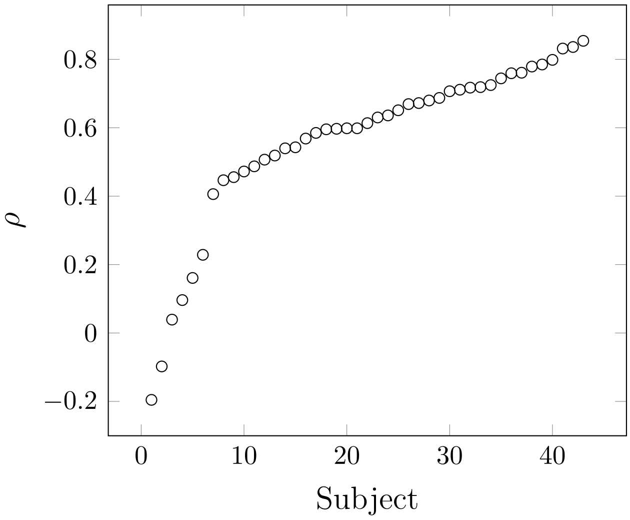

Figure 3.5: Sorted Spearman correlation coefficients between SenticNet and human rankings

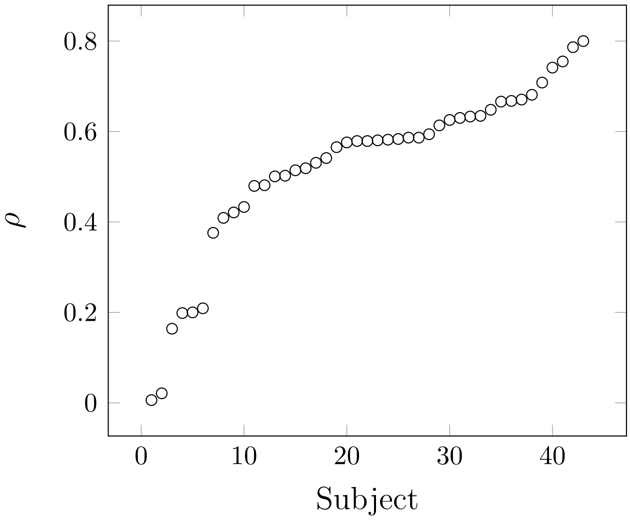

Figure 3.6: Sorted Spearman correlation coefficients between SentiWordNet and human rankings

Chapter 4

Movie Review Analysis

In this chapter, our purpose is to assign scores to the Turkish movie reviews using adjective clusters. Firstly, we explain how to install and use the TRmorph which is a freely available open-source morphological analyzer tool that helps us for POS tagging for selection of adjectives in movie reviews. After installation and usage of this tool, we talk about data preparation. Then, we give our methodology that is adjective clustering with training and testing stages. At the end of these stages, we calculate error rates. In addition, we do this calculation on another methodology which is singleton review scores. Finally, we give our results and evaluations.

4.1 TRmorph - A Morphological Analyzer for Turkish

TRmorph is a freely available open-source morphological analyzer for Turkish. It provides morphological segmentation, stemming, lemmatization, guessing unknown words, grapheme to phoneme conversion, hyphenation and morphological disambiguation tools [11].

4.1.1 Installation

In this part, we talk about installation of TRmorph on Ubuntu 14.04. Firstly, we can download TRmorph from `http://github.com/coltekin/TRmorph` because it is a free and distributed tool with a license that allows free use. Also, we download all dependencies which are “foma” and “flookup” that are parts of foma which is a finite state compiler [12]. To download the dependencies, we can use currently available repository that is on the `https://bitbucket.org/mhulden/foma/downloads`.

In order to use foma and flookup for batch processing, we write the following commands on the terminal.

```
$ cp foma /usr/local/bin
$ cp flookup /usr/local/bin
```

Then, we write “foma -f analyzer.cpp.xfst” command on terminal under the TRmorph folder to create “trmorph.fst” which is a binary transducer in foma format.

4.1.2 Trying it out with POS Tags

Part of speech tags indicate the category of the words with their syntactic functions. All part of speech (POS) tags used in TRmorph include “Alpha”, “Adj”, “Adv”, “Cnj”, “Det”, “Exist”, “Ij”, “N”, “Not”, “Num”, “Onom”, “Postp”, “Prn”, “Punc”, “Q”, “V” and listed in Table 4.1 with their descriptions.

As a last step, we run the following example command and see the results as shown in following.

```
$ echo "okudum" |flookup trmorph.fst
okudu   oku<v><past><1s>
```

4.2 Data Preparation

For data preparation stage, firstly we select the Milliyet corpus [13] to extract adjectives that we use in clustering. Then, we apply graph clustering to generate the clusters on the adjectives. Moreover, we collect movie reviews from internet

Tag	Description
Alpha	Symbols of the alphabet
Adj	Adjective
Adv	Adverb
Cnj	Conjunction
Det	Determiner
Exist	The words var and yok
Ij	Interjection
N	Noun
Not	The word değil
Num	Number
Onom	Onomatopoeia
Postp	Postposition
Prn	Pronoun
Pun	Punctuation
Q	Question particle mI
V	Verb

Table 4.1: All part of speech tags used in TRmorph

for training and test stages. Normalization is applied on movie review dataset for deasciification, vowel restoration, accent normalization and spelling correction.

4.2.1 Corpus

In order to generate clusters with adjectives, it is necessary to extract adjectives from a large Turkish Corpus. We selected Milliyet Corpus [13] which consists of three parts. Each part has sentences that are divided into lines word by word as shown in below. $< S >$ means that sentence starts from here, also $< /S >$ means that sentence ends there. In below, there are two sentences which are; "Kuşkusuz bu çalışma onları belki de yılın en karlı gecesini yaşattı" and "Yılbaşı'nda program yapan ünlüler arasında en yüksek ücreti İbrahim Tatlıses aldı".

Because of large size of Corpus file, we splitted the Corpus file into small size of files. After that, we applied TRmorph on each small file in order to get their morphological analysis. We saved all analysis in different files. When we consider the results of analysis, we saw that there are unnecessary datas in analysis. To

```
< S >
Kuşkusuz
bu
çalışma
onları
belki
de
yılın
en
karlı
gecesini
yaşattı
< /S >
< S >
Yılbaşı'nda
program
yapan
ünlüler
arasında
en
yüksek
ücreti
İbrahim
Tatlıses
aldı
< /S >
```

remove these unnecessary datas, we did data reduction. In data reduction, we stated some rules with details below.

- "typo" Reduction Rule

  For this reduction rule, we eliminated the lines that contains ":typo" in analysis because of their root form. When we analyse a word for example; "bilim". TRmorph gives an analysis that starts with "bir" as shown in below. Consequently, we just take the analysis that starts with relevant word.

  ```
  bilim    bir<Num:typo><0><N><li><Adv><0><N><p1s>
  bilim    bir<Num:typo><0><N><li><Adv><0><N><p1s><0><V>
  ```

```
bilim    bir<Num:typo><0><N><li><Adj><0><N><p1s>
bilim    bir<Num:typo><0><N><li><Adj><0><N><p1s><0><V>
bilim    bilim<N>
bilim    bilim<N><0><V>
bilim    bilim<N><0><V><cpl:pres>
bilim    bilim<N><0><V><cpl:pres><3s>
```

- “mredup” Reduction Rule

  In this reduction rule, we removed the lines that contains “:mredup” due to the fact that the root form of these analysis is different than the word that we have analysed such as; our word is “mark” and the results in following. We are just interested in analysis of our word, not the analysis of similar words.

```
mark    mark<N>
mark    mark<N><0><V>
mark    cark<N:mredup>
mark    cark<N:mredup><0><V>
mark    Sark<N:prop:mredup>
mark    Sark<N:prop:mredup><0><V>
mark    park<N:mredup>
mark    park<N:mredup><0><V>
```

- Similar Reduction Rule

  In this rule, we found similar results of analysis, then we removed unrelevant ones. If one line analysis starts with any other lines, we remove this line. In below example, lines except “bilim bilim$< N >$” line starts with “bilim bilim$< N >$”, therefore we remove all other lines. We just take the line “bilim bilim$< N >$” as a result of analysis.

```
bilim    bilim<N>
bilim    bilim<N><0><V>
bilim    bilim<N><0><V><cpl:pres>
bilim    bilim<N><0><V><cpl:pres><3s>
```

- Accent Mark Reduction Rule

  In this reduction rule, we use replacing on the characters that has an accent mark replacing with the same character without accent mark. The analysis of word “mekan” is shown in below. Some lines contains this word with accent mark. So, we change these to normal characters.

```
mekan    mekan<N>
mekan    mekan<N><0><V>
mekan    mekan<N><0><V><cpl:pres>
mekan    mekan<N><0><V><cpl:pres><3s>
mekan    mekân<N>
mekan    mekân<N><0><V>
mekan    mekân<N><0><V><cpl:pres>
mekan    mekân<N><0><V><cpl:pres><3s>
```

- “cpl:pres” Reduction Rule, “la suffix” Reduction Rule and “partial” Reduction Rule

  The combination of these three rule is to eliminate the lines that contains either “cpl:pres” or “:partial” in the analysis. If any lines contains “la suffix”, we also eliminate that line.

- Selection Rule

  In this selection rule, if there exists an analysis that indicates the word as both adjective and adverb of a given word, we select the Adjective and eliminate Adverb from analysis. In addition, if the analysis indicates the word as both Adjective and Noun, we chose Noun. In below, there is an example which has two types of POS tags in analysis, so we chose “güzel” as Adjective with our selection rule.

```
güzel    güzel<Adv>
güzel    güzel<Adj>
```

After data reduction stage, we did POS tagging on reduced and analyzed datas. In order to apply POS tagging, we consider two important rules of analysis as given in following:

- The first one is to check the all root words of the whole analysis of the word are the same.
- The second one is to check all last tags of the whole analysis of the word are the same.

According to these rules, we give POS tags to the words in the Corpus. If any word is not appropriate for the rules, given POS tag is $< UNK >$ which means unknown. Otherwise, we give POS tag of it from the list of POS tags.

We based our method on the intuition that two adjectives joined by "and" have the same polarity [14]. We use this to cluster similar adjectives. After POS tagging, we extracted the adjectives which are joined by "and" with respect to our methodology given in Table 4.2.

Adjectives joined by "and"
demokratik ve laik
samimi ve iyi
kirli ve nemli
yatay ve dikey
tropikal ve ılık

Table 4.2: Samples for extracted adjectives joined by "and"

4.2.2 Graph Clustering

Using the extracted adjectives joined by "and", we want to generate clusters on a graph. In order to generate clusters that include adjectives on a graph, we need to create a graph with nodes that contains adjectives as data. Therefore, we used NetworkX which is a Python library for the creation of networks [15]. With the help of this library, we built this graph. In this graph, adjectives joined by "and" were nodes with edges between them.

Example 4.2.1. Assume that we have a list that contains adjectives joined by "and" such as; "kirli ve nemli", "tropikal ve ılık", "iyi ve hoş" and "güzel ve iyi". If we build a graph with these adjectives, it will be like Figure 4.1.

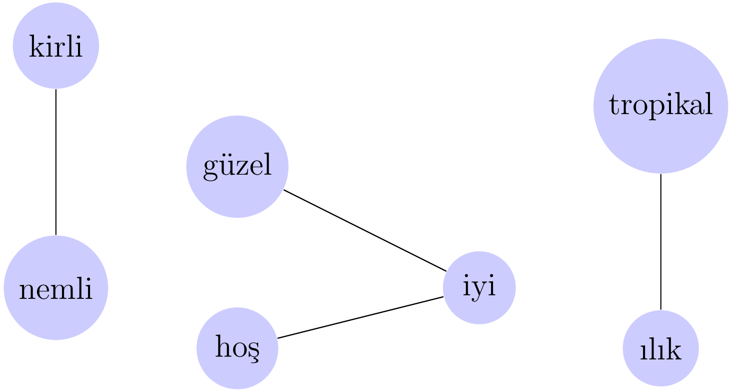

Figure 4.1: An example of joined adjectives as connected nodes

After graph building, next step is to generate clusters. For the clusters generating, we used the Louvain method described in [16]. Louvain method is a simple and efficient method to extract communities in large networks. This method is based on modularity optimization that looks firstly for small communities by optimizing modularity and finds the same communities. For the usage of Louvain method on NetworkX graph, there is a community detection module on `https://bitbucket.org/taynaud/python-louvain`. By using Louvain method of this community detection module in python with NetworkX, we generated our adjective clusters. Thus, adjectives that have same polarity are assumed to fall in the same clusters with respect to our method. As a result, 260 adjective clusters are generated with 2313 adjectives. Then, we stored them in a JSON file with their adjectives and initial score of zero.

4.2.3 Movie Review Data

For movie review analysis, we need movie review dataset. Consequently, we searched an online website that contains many Turkish movie reviews, and we found `http://www.beyazperde.com`. In this website, there are a lot of reviews on various type of movies. For each review, there are six options which range from 0 to 5 stars to give score. In order to collect reviews from this website, we manually got all needed datas that contain reviews and scores. To set the dataset uniformly distributed, we collected 100 reviews equally from all types of scores. Then, we saved our dataset as a JSON format which is easy to read and write.

4.2.4 Normalization

People generally write reviews or comments with a somewhat broken grammar on the social media. Because of this, the data that are collected from internet contains many spelling mistakes. Therefore, we need to apply preprocessing on the movie review dataset to correct these spelling mistakes and others. In preprocessing step, we have applied normalization on our dataset. For Turkish social media texts, there is a NLP tool which is "ITU Turkish NLP Web Service" that provides a normalizer [17]. Using this web service, we normalized our dataset. In Table 4.3, there is an example from our dataset and normalized dataset.

Review	**Normalized Review**
süperrrrrrrr,ona reddedemeyeceği bir teklif yapacağım	Süper ona reddedemeyeceği bir teklif yapacağım
gerçekten çok komik olmuş ya bayağı bi güldm; ama, izlenmese de olur bence, bu hafta daha ii filmler war	Gerçekten çok komik olmuş ya bayağı bir güldüm ama izlenmese de olur bence bu hafta daha iyi filmler var

Table 4.3: A part of reviews and normalized reviews

4.2.5 Morphological Analysis

After we have applied normalization on movie reviews, we obtain a more clear movie review dataset. In our clusters that we have generated using graph clustering, all adjectives are in root form. That's why we need POS tags of each word of movie reviews in order to find adjectives and their root forms. By using TRmorph we have applied POS tagging on our normalized movie review dataset. Then, we stored POS tagged version of each movie review in a JSON file again. Now, we can apply training and test steps on this dataset easily.

4.3 Training and Testing using Adjective Clusters

For training part, we assign scores to each cluster using movie reviews. To begin with, we assumed that each cluster that we have generated has a score. Initially scores of all clusters are zero. In order to assign new scores for these clusters, we decided to use our movie reviews. Each movie review sentence has a score in a structured JSON file. Therefore, we proposed a training method that uses equation 4.1.

$$C_i = \frac{\sum\limits_{j=1}^{N_i}\sum\limits_{z=1}^{M} a_{jz}R_z}{\sum\limits_{j=1}^{N_i}\sum\limits_{z=1}^{M} a_{jz}} \tag{4.1}$$

In equation 4.1, C_i is score of cluster i, N_i is number of adjectives that belong to cluster i, M is the number of reviews, R_z is the z^{th} review score and a_{jz} is number of j^{th} adjectives that occur in z^{th} review.

In our training method, we suppose C is set of clusters and R is set of reviews. In order to assign a score to a cluster, we firstly get all adjectives that belong to that cluster. For each adjective of that cluster, we look at the number of occurrences of adjective in each review. Then, we multiply the number of occurrences of each adjective in a review with score of review that adjective occurs. Here is a small example to illustrate the calculation.

Example 4.3.1. Assume C_1 is the cluster that we would like to assign a new score, a_1 and a_2 are adjectives of this cluster and R_1, R_2 and R_3 are the reviews in our movie review dataset. Suppose that scores of reviews are $R_1 = 2$, $R_2 = 4$ and $R_3 = 5$. Also, a_1 occurs 1 time in R_1, 3 times in R_3 and a_2 occurs 2 times in R_2, 1 time in R_3 as shown in Table 4.4.

	R_1	R_2	R_3
a_1	1	0	3
a_2	0	2	1

Table 4.4: Small example for cluster training

We compute score of C_1 as follows using equation 4.1.

$$\begin{aligned}
C_1 &= \frac{\sum\limits_{j=1}^{2}\sum\limits_{z=1}^{3} a_{jz}R_z}{\sum\limits_{j=1}^{2}\sum\limits_{z=1}^{3} a_{jz}} \\
&= \frac{(a_{11}R_1) + (a_{12}R_2) + (a_{13}R_3) + (a_{21}R_1) + (a_{22}R_2) + (a_{23}R_3)}{a_{11} + a_{12} + a_{13} + a_{21} + a_{22} + a_{23}} \\
&= \frac{(1 \times 2) + (0 \times 4) + (3 \times 5) + (0 \times 2) + (2 \times 4) + (1 \times 5)}{1 + 0 + 3 + 0 + 2 + 1} \\
&= \frac{(2) + (0) + (15) + (0) + (8) + (5)}{7} \\
&= \frac{30}{7} \\
&= 4.285
\end{aligned}$$

As we see from this example, score of C_1 cluster is 4.285 after training. For each cluster we apply these steps.

For testing part, our purpose is to predict the scores of given movie reviews. Firstly, we collected movie reviews without their scores. They have been saved to JSON file as we did in training movie review dataset. Then, we used the trained clusters to give a score to each test movie review. Bu using TRmorph morphological analyzer, we found POS tags of our review sentences. From these tagged words of review sentences, we have just taken adjectives for calculation. Using our testing method with equation 4.2, we found scores of each test movie review.

$$R_i = \frac{\sum\limits_{a \in A} |a^i| C(a)}{\sum\limits_{a \in A} |a^i|} \tag{4.2}$$

In equation 4.2, R_i is score of review i, A is the set of adjectives that occur in R_i, $|a^i|$ is the number of occurrence of adjective a belongs to the set A in R_i. $C(a)$ is the cluster score of given adjective a.

In our testing method, we suppose R is set of reviews. First of all, we get all adjectives of each reviews by using TRmorph. We assume that all adjectives that we got from each review belongs to the set A. For each elements of this set we find number of occurrences of them in each review. Then, we find their cluster. By multiplying number of occurrences of adjectives and their cluster score, we calculate a summation. Lastly, we divide this calculation to sum of number of occurrences of each adjectives that are in A. Here is a small example to illustrate.

Example 4.3.2. Assume that R_1 is the review that we would like to predict score of it, a_1, a_2 and a_3 are adjectives that occur one time for each in this review. $a_1 \in C_2$, $a_2 \in C_5$ and $a_3 \in C_{10}$ where the scores of clusters $C_2 = 4.8$, $C_5 = 3.2$ and $C_{10} = 3.9$.

We compute score of R_1 as follows using equation 4.2.

$$
\begin{aligned}
R_1 &= \frac{\sum\limits_{a \in A} |a^1| C(a)}{\sum\limits_{a \in A} |a^1|} \\
&= \frac{(|a_1^1| C(a_1)) + (|a_2^1| C(a_2)) + (|a_3^1| C(a_3))}{|a_1^1| + |a_2^1| + |a_3^1|} \\
&= \frac{(|a_1^1| C_2) + (|a_2^1| C_5) + (|a_3^1| C_{10})}{|a_1^1| + |a_2^1| + |a_3^1|} \\
&= \frac{(1 \times 4.8) + (1 \times 3.2) + (1 \times 3.9)}{1 + 1 + 1} \\
&= \frac{11.9}{3} \\
&= 3.96
\end{aligned}
$$

As we see from this example, score of R_1 review is 3.96 after testing. For each review, we apply these steps.

4.4 Error Calculation

In order to calculate the errors for training and test stages, we have used the equation 4.3.

$$
Error = \frac{\sum\limits_{i=1}^{C_{ReviewCount}} \frac{|\widehat{S}_{Ri} - S_{Ri}|}{MaxScore}}{C_{ReviewCount}} \tag{4.3}
$$

In equation 4.3, $C_{ReviewCount}$ is number of reviews, $\widehat{S}_{Ri}$ is actual score of review i, S_{Ri} is predicted score of review i and $MaxScore$ is 5 as constant because our range is in [0, 5] for review scores.

4.5 Cross Validation

In this chapter, our purpose is to predict the movie review scores. So, we have collected movie review datas. However, we have limited amount of data. Because of limited amount of data, we used splitting with cross validation. In cross validation, a part of data is used for training and rest of data is used for testing part. In generally, we split data to %90 training and %10 test. We call this 10-fold cross validation. Fold means the splitting size of cross validation procedure.

For our experiment, we have 600 movie reviews. Therefore, we splitted our data to 540 training samples and 60 testing samples. In order to apply 10-fold cross validation, for each folding part, we choose different data for training and testing. Then we calculate the errors for each time. There are 10 error calculations at the end. Then, we use equation 4.4 in order to find the average of these errors.

$$E = \frac{1}{K} \sum_{i=1}^{K} E_i \tag{4.4}$$

where E is the average of errors from cross validation, K is the number of folds and E_i is the error rate of each ith training step.

4.6 Singleton Review Scores

In graph clustering part, we have created a graph by using the adjectives joined by "and". Then, we had various clusters via this graph. Moreover, we assigned scores for each cluster using the reviews in training stage. For test stage, we predicted the scores of reviews score assigned clusters.

In this part, we have considered adjectives as singleton in terms of their scores rather than using them in a cluster. Therefore, we have assigned scores to each adjective by looking their frequencies in each review. Then, we predicted scores of test reviews using these score assigned adjectives.

In order to assign scores to adjectives by looking their frequencies in all movie reviews using their scores, we use the equation 4.5.

$$a_i = \frac{\sum\limits_{j=1}^{N} C(R_j) R_j}{\sum\limits_{j=1}^{N} C(R_j)} \tag{4.5}$$

In equation 4.5, a_i is score of adjective i, N is the number of reviews, R_j is the j^{th} review score and $C(R_j)$ is the number of occurences of adjective a_i in review R_j.

For movie review score prediction, we use score assigned adjectives on test movie reviews using equation 4.6.

$$R_i = \frac{\sum\limits_{a \in A} |a^i| S(a)}{\sum\limits_{a \in A} |a^i|} \tag{4.6}$$

In equation 4.6, R_i is score of review i, A is the set of adjectives that occur in R_i, $|a^i|$ is the number of occurrence of adjective a belongs to set A in R_i. $S(a)$ is the score of score assigned adjective a.

4.7 Results and Evaluation

In the movie review analysis chapter, we performed two types of score assignments. The first one is based on adjective clusters and the other one is singleton review scores. We applied 10-fold cross validation in both stages on 600 movie reviews. For adjective clustering, accuracy rate is 0.76 and the error rate is 0.24 with 0.01 standard deviation. For the singleton review scores, the accuracy is 0.79 and the error rate is 0.21 with 0.02 standard deviation as given in Table 4.5.

Method	**Accuracy Rate**	**Error Rate**	**Std Deviation**
Adjective Clustering	0.76	0.24	0.01
Singleton Review Scores	0.79	0.21	0.02

Table 4.5: Accuracy and error rates of score assignments

Chapter 5

Discussion

In this chapter, we discuss the basic reasons of unexpected results of experiment which measures the agreement between the polarities of sentiment lexicons and human judgments. Also, we give examples that cause bad results for our movie review analysis.

Before we conduct an experiment that compares two popular sentiment lexicons to human judgments for English adjectives, we conducted experiments for Turkish adjectives. We selected 20 most frequently used adjectives from Milliyet Corpus. For the annotator pool, we chose 40 students from Işık University. They applied pair comparison with these 20 Turkish adjectives using Binary Insertion Sort as we did in Chapter 3. Then, we compared the ranking results of annotators with two ranking methods that we defined. These are "Translated Ranking Method" and "Average Translated Ranking Method".

For "Translated Ranking Method", we sorted the Turkish adjectives in terms of SenticNet polarities of their direct English translations using Google Translate. For "Average Translated Ranking Method", we first translated each adjective to English. Google Translate serves multiple translations for many words with their weights. Then, we calculated new polarities for each Turkish adjective with the help of these weights and SenticNet polarities of each adjective. Using the Spearman correlation coefficients between human rankings and these two methods, we evaluated the correlations between them. Correlations were not good and as expected. Poor results could be due to translation of adjectives. We have changed our study to English.

The studies for this thesis showed that there is a moderate level agreement between sentiment lexicons and human judgments. Annotator pool was small for both studies and we chose them from spesific areas, therefore annotator pool could be expanded. We have published our results in a conference paper about validity assessment of prior polarities in sentiment lexicons [18].

For movie review analysis, we started with graph clustering. Firstly, we started with an NP which consists of at least one Adjective and a Noun. Then, we extracted these NP's such as; "sert-yıpratıcı-markaj", "iyi-konu" from Milliyet Corpus. We connected the adjectives with respect to the nouns they qualify. If two different adjectives have same noun, they are connected. For example; "iyi-adam" and "düzgün-adam". From these two NP's, we can say that "iyi" and "düzgün" are connected. Before graph clustering, we observed that many irrelevant adjectives are connected to each other. From the NP's which are "iyi film", "kötü film", we can see that "iyi" and "kötü" adjectives are opposite adjectives, and they do not need a connection. Therefore, we have changed our approach as follows.

We used an approach based on adjectives joined by "and" for graph clustering in movie review analysis. Results and evaluations show that accuracies are under our expectations to predict a score of any movie review. In some clusters, some adjectives are semantically distant from other words in those clusters. For example; the adjective "yıldırıcı" appeared in a cluster that contains adjectives with a positive meaning. However, "yıldırıcı" adjective could be in an another cluster which has negative adjectives. This is contradiction, and this originates from "etkili ve yıldırıcı" pair in Milliyet Corpus. Moreover, we did not have a disambiguator for word sense of adjectives. This reduced the number of NP's that we could use to only the set where the modified set of analyses contains a single chain of morphological analyses. If we had used a disambiguator, we expect to use a considerably larger set of NP's with connected adjectives.

Chapter 6

Conclusion

In this thesis, we designed and implemented an experiment to determine the agreement between two important sentiment lexicons and human polarity judgments to start with a sentiment lexicon for sentiment analysis. We also performed graph clustering task based on adjectives in order to predict movie review scores.

In chapter 3, we showed the results of experiment which is done between human rankings and two popular sentiment lexicons, SenticNet and SentiWordNet. We performed human judgment with selected terms to assess the validity of 20 sentiment scores from these two sentiment lexicons. The correlation between human ranked polarities and lexicon polarities is not high enough. The results show that there is a moderate level of agreement between human judgment and sentiment lexicons. In further studies, the number of the human subjects can be increased.

In chapter 4, we gave the results of adjective clustering and singleton review scores for movie review score prediction. As we see from their results, there is a small difference between these two methodologies. Also, results are not sufficiently accurate, therefore the methods are not applicable for any movie review score prediction system. In future, more movie reviews can be used for cluster training. Besides, adjective clustering can be applied on English movie reviews in order to see the differences between languages.

References

[1] B. Liu, *Sentiment analysis: Mining opinions, sentiments, and emotions.* Cambridge University Press, 1 2015.

[2] C. Dalitz and K. E. Bednarek, "Sentiment lexica from paired comparisons," in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Dec 2016, pp. 924–930.

[3] N. Godbole, M. Srinivasaiah, and S. Skiena, "S.: Large-scale sentiment analysis for news and blogs," in *In: Proc. Int. Conf. Weblogs and Social Media (ICWSM)*, 2007.

[4] D. Okanohara and J. Tsujii, "Assigning polarity scores to reviews using machine learning techniques," in *International Conference on Natural Language Processing.* Springer, 2005, pp. 314–325.

[5] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *LREC*, N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, Eds. European Language Resources Association, 2010.

[6] E. Cambria, S. Poria, R. Bajpai, and B. W. Schuller, "Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives." in *COLING*, N. Calzolari, Y. Matsumoto, and R. Prasad, Eds. ACL, 2016, pp. 2666–2677.

[7] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995. [Online]. Available: http://doi.acm.org/10.1145/219717.219748

[8] T. H. Cormen, *Introduction to algorithms.* MIT press, 2009.

[9] W. N. Francis and H. Kucera, "Brown corpus manual," Brown University, Tech. Rep., 1979.

[10] C. Spearman, "The proof and measurement of association between two things," *The American Journal of Psychology*, vol. 15, no. 1, pp. 72–101, 1904.

[11] C. Çöltekin, "A set of open source tools for Turkish natural language processing," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds. European Language Resources Association (ELRA), 2014, pp. 1079–1086. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2014/pdf/437_Paper.pdf

[12] M. Hulden, "Foma: a finite-state compiler and library," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics.* Association for Computational Linguistics, 2009, pp. 29–32.

[13] H. Sak, T. Güngör, and M. Saraclar, "Turkish language resources: Morphological parser, morphological disambiguator and web corpus," in *GoTAL*, 2008.

[14] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ser. ACL '98. Stroudsburg, PA, USA: Association for Computational

Linguistics, 1997, pp. 174–181. [Online]. Available: https://doi.org/10.3115/976909.979640

[15] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Pasadena, CA USA, Aug. 2008, pp. 11–15.

[16] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008. [Online]. Available: http://stacks.iop.org/1742-5468/2008/i=10/a=P10008

[17] G. Eryiğit, "ITU Turkish NLP web service," in *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Gothenburg, Sweden: Association for Computational Linguistics, April 2014.

[18] A. B. Kanburoğlu and E. Solak, "An experimental evaluation of prior polarities in sentiment lexicons," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, Nov 2017, pp. 389–392.