WORD POLARITY DETECTION USING A MULTILINGUAL APPROACH

by

Cüneyd Murad Özsert

B.S., Computer Engineering, Boğaziçi University, 2008

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2012

ACKNOWLEDGEMENTS

Thanks to my adviser, Arzucan Özgür, for all of her support, patience and help on preperation of this work. I also thank Suzan Üsküdarlı and Şule Gündüz Öğüdücü for participating in my thesis jury and giving me feedback.

I also thank Tuna Tuğcu, for his valuable guidance in my master education.

I would like to express my great feelings to my friends, especially Elif, who always motivate me to complete my thesis. I can not forget your support throughout my life.

I can not forget Vodafone, which supported me to complete my master education. I would like to thank my managers, Osman Yaycıoğlu, Aslı Emek and Çağatay Tunalı who motivated and supported me in all phases of this work.

Thanks to Amjad Abu-Jbara and Ahmed Hassan from the University of Michigan. They always helped me when I got stuck during my thesis preperation.

I would like to thank my chair which became a good friend for long lasting nights.

Thanks to my family, Sevde, Mustafa, Hülya, Gökçenas especially my parents, Semra and İbrahim Özsert, for everything. I can not forget your support and love throughout my life.

Thanks to all that I haven't met yet and will contribute to my life.

ABSTRACT

WORD POLARITY DETECTION USING A MULTILINGUAL APPROACH

Determining polarity of words is an important task in sentiment analysis with applications in several areas such as text categorization and review analysis. In this thesis, we propose a multilingual approach for word polarity detection. We construct a word relatedness graph by using the relations in WordNet of a given language. We extend the graph by connecting the WordNets of different languages with the help of the Inter-Lingual-Index based on English WordNet. We develop a semi-automated procedure to produce a set of positive and negative seed words for foreign languages by using a set of English seed words. In our approach, these seed words are used for semi-supervised learning and for evaluation purposes. To identify the polarity of unlabeled words, we propose a method based on random walk model with commute time metric as proximity measure. We evaluate our multilingual approach for English and Turkish and show that it leads to improvement in performance for both languages. To the best of our knowledge, we report the first word polarity detection results for Turkish.

ÖZET

ÇOKLU DİL YAKLAŞIMI İLE KELİMELERİN ANLAMSAL YÖNELİMİNİ BULMA

Kelimelerin anlamsal yönelimini belirleme, duygu analizinde, metin sınıflandırma ve yorum analizi gibi uygulamalarda kullanılan önemli bir konudur. Biz bu çalışmada, kelimelerin anlamsal yönelimini bulmak için çok dilli bir yaklaşım öneriyoruz. Bu yaklaşımda, ilgili dil için WordNet'te bulunan kelimeler arası ilişkileri kullanarak, kelimelerin ilişkisini gösteren bir ağ yapılandırıyoruz. Daha sonra bu ağı farklı dillerin WordNet'lerini bağlayarak genişletiyoruz. Bunu yapmak için İngilizce Wordnet baz alınarak geliştirilen diller arası indeks denilen kavramı kullanıyoruz. Ayrıca yabancı diller için önceden anlamsal yönelimi işaretli (positif ve negatif) kelimeleri üretmek için de yarı otomatik bir işaretleme yöntemi öneriyoruz. Bu yöntemde önceden işaretli İngilizce kelimelerden faydalanıyoruz. Önceden anlamsal yönelimi işaretli bu kelimeleri makina öğrenmesinde ve performans değerlendirmesinde kullanıyoruz. İşareti bilinmeyen bir kelimenim işaretini belirlemek için, bu ağ üzerinde rastlantısal yürüyüş modelini uyguluyoruz. Yakınlık ölçüsü olarak gidiş-geliş metriğini kullanıyoruz. Bizim çok dilli yaklaşımımızı Türkçe ve İngilizce için test ediyoruz ve iki dil için de tek dilli yaklaşımlara göre daha iyi bir performans gerçekleştiğini gösteriyoruz. Bildiğimiz kadarıyla bu çalışma Türkçe için kelimelerin anlamsal yönelimini belirleme alanındaki ilk çalışmadır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

c_{ij}	the commute time between vertices i and j
c_{ij}^T	the truncated commute time between vertices i and j
h_{ij}	the hitting time between vertices i and j
h_{ij}^T	the truncated hitting time between vertices i and j
N	the set of negative seed words
M	the number of random walks
p_{ij}	the probability of moving from vertex i to vertex j
P	the set of positive seed words
T	the maximum length of a random walk
W_{ij}	the weight of the edge between vertices i and j

LIST OF ACRONYMS/ABBREVIATIONS

ILI	Inter-Lingual-Index

1. INTRODUCTION

The evaluative character of a word is called its semantic orientation or polarity [1]. Positive semantic orientation indicates praise (e.g., "good", "honest"), whereas negative semantic orientation indicates criticism (e.g., "bad", "disturbing"). In addition, the semantic orientation or polarity of a word is also defined as the direction the word deviates from the norm for its semantic group or lexical field [2].

Identifying polarity of a word is one of the most important topics in sentiment analysis. Many application areas are based on the polarity of individual words. For instance, consider the task of analyzing product reviews [3, 4]. According to the results of product review analysis, companies may focus on the reviews that are not positive and change their strategies related for the associated products accordingly. In addition, consumers can also use the results of product review analysis when deciding which product to buy. Given the vast amount of reviews available for several products, an automated process for classifying product reviews is needed. Identifying the polarities of the words in product reviews would be very helpful for such an automated method.

Another important application of sentiment analysis, which can benefit from word polarity detection, is classifying movie reviews as recommended or not recommended. For example, in [5], Turney extracts adjective phrases from each review and determines the polarity of each individual phrase. Then, the average semantic orientation of the phrases is used for classifying movie reviews. In other words, polarity of individual words is used to identify whether a movie is recommended or not. Determining the attitudes of participants in online discussions is another interesting example that makes use of word polarity detection [6].

With the rapid development of the Internet, social media became an important platform for comments and reviews about everything. Every event triggers many contributions by social media users. If we can analyze user contributions in realtime, useful information that can be used by companies, politicians and other organizations can be

obtained. For instance, when a company launches a campaign about a new product, social media users immediately comment about this campaign. If we can classify user feedbacks in realtime, marketing departments can analyze them and manage their campaigns more successfully. Identifying the polarity of each word in the feedback would be very helpful for analyzing these feedbacks.

Most previous studies on word polarity detection have been carried out for English and make use of language-specific resources such as WordNet [7] and General Inquirer [8]. Wordnet is a large lexical database for English, consisting of synsets (i.e. set of synonyms) each associated with a distinct meaning [7]. Each synset consists of words with the same meaning. In addition, General Inquirer is a computer-assisted approach for content analysis of textual data [8]. It consists of 182 tagged categories such as positive and negative. In polarity detection studies WordNet has mainly been used to construct word relatedness graphs by connecting semantically related words and General Inquirer has been used to obtain seed words labeled as positive or negative for supervised/semi-supervised machine learning settings and for evaluation purposes [9, 10]. Many languages do not have semantically tagged lexicons such as General Inquirer. Even though some of these languages have WordNets, they are in general not as comprehensive as the English WordNet.

Most Non-English WordNets such as EuroWordNet [11] and BalkaNet [12, 13] are structured in the same way as English WordNet [7] in terms of synsets and basic relations between synsets. As seen in Figure 1.1, WordNets are linked to each other with Inter-Lingual-Index (ILI) based on English WordNet. With the help of this index, it is possible to reach from a synset in any language to the synsets of the same meaning in any other language. As seen in Figure 1.1, the word "drive" in English can directly be connected to its similar meaning in Spanish, Dutch and Italian by using ILI.

In this thesis, we take advantage of the compatibility of WordNets and develop a multilingual approach for detecting polarities of English as well as Non-English words. We construct a word-relatedness graph by not only connecting semantically related words in one WordNet but by also linking words from WordNets of different languages.

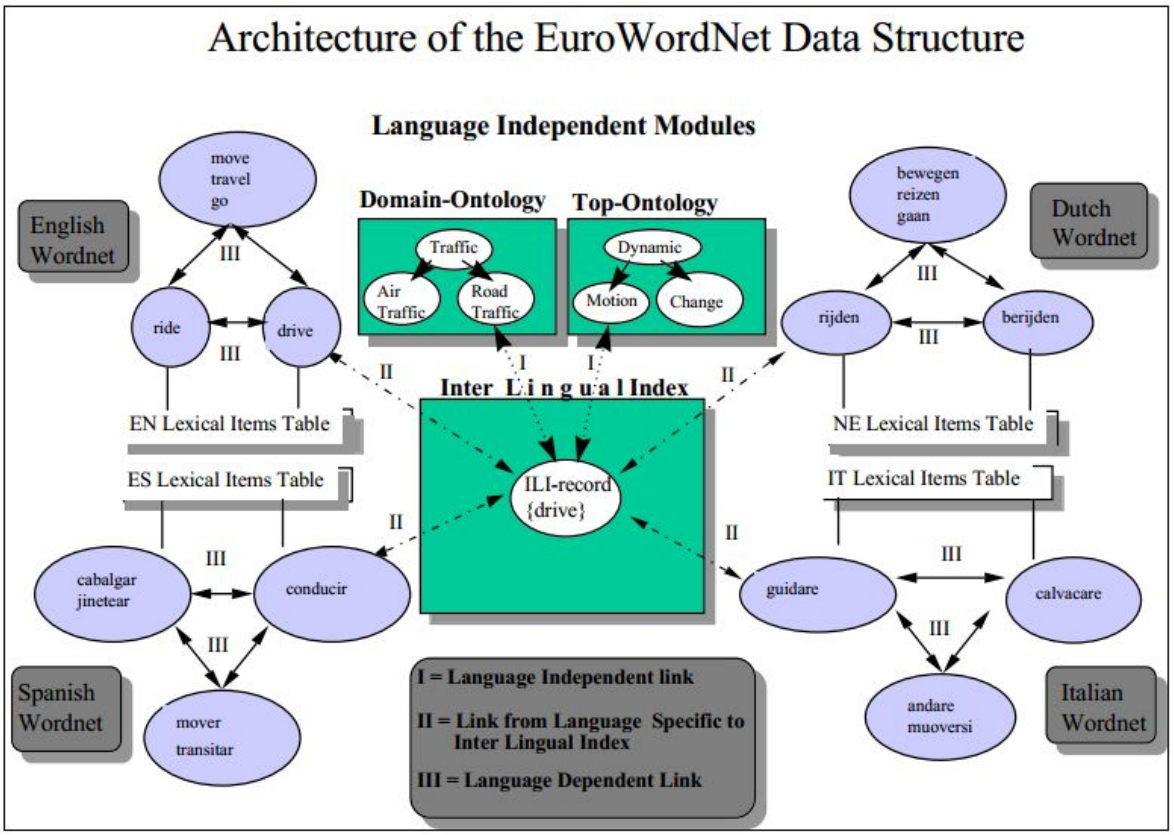


Figure 1.1. WordNets are linked to each other with ILI based on English WordNet[1].

We also propose a semi-automated method to generate labeled seed words for other languages by using the list of English seed words and the ILI. Then, we define a random walk over the word-relatedness graph from any given word to the set of positive and negative seed words. We use commute time as a proximity measure and classify a given word as positive if it is closer to the set of positive seed words compared to the negative seed words, and classify it as negative otherwise. We evaluate our approach for English and Turkish. Turkish WordNet [14, 15] is completed within the BalkaNet project [12]. It is constructed as being fully compatible with EuroWordNet, which in turn is compatible with English WordNet. We first show that our commute time model achieves performance comparable to the state-of-the-art in the literature. Then, we demonstrate that creating a multilingual word relatedness graph by connecting the WordNets of English and Turkish boosted the performance of word polarity detection for both languages.

[1]This figure is taken from EuroWordNet General Document, July 1, 2002.

To our knowledge, we report the first results for Turkish word polarity detection and achieve an accuracy of 95.5%. Our approach makes use of the relatively rich English linguistic resources and can also be applied to other languages that have WordNets compatible with the English WordNet. We also believe that linking WordNets to each other by using ILI opens a powerful vision to other researchers and provides them to take advantage of rich English linguistic resources.

The outline of this thesis is as follows. In Chapter 2 we present background information about word polarity detection. We summarize related work in Chapter 3. Our approach is explained in Chapter 4. In Chapter 5, our method is evaluated by comparisons of monolingual and multilingual experiments. Our conclusions and future work are presented in Chapter 6 and Chapter 7, respectively.

2. BACKGROUND

In this chapter, we provide background about the concepts and resources used in this thesis.

2.1. English WordNet

WordNet, which is developed by Princeton University, is a large lexical database of English words organized as set of synonyms (synsets) [7]. Each of these synsets expresses a distinct meaning. Synsets are linked to each other by several semantic relations. As shown in Table 2.1, WordNet 2.0 has 115424 synsets. It has also 203145 word-sense pairs. Each word-sense pair identifies a specific sense of a word. For example, "find-9" means making discovery whereas "find-4" means determining. The number of word-sense pairs is larger than the number of synsets because each synset consists of one or more word-sense pairs.

Table 2.1. English WordNet statistics.

POS	**Unique Strings**	**Synsets**	**Total Word-Sense Pairs**
Noun	114648	79689	141690
Verb	11306	13508	24632
Adjective	21436	18563	31015
Adverb	4669	3664	5808
Totals	152059	115424	203145

The WordNet database and tools can be downloaded and used freely from the WordNet website[2]. We use WordNet 2.0 in this study since it is compatible with Turkish WordNet [14]. We also use the XML Structure of English WordNet. WordNet groups sets of words together according to their meanings. As a result, if we construct a word relatedness graph by using the semantic relations between words, words that are close to each other can be considered semantically related. In this thesis, we make

[2]The website wordnet.princeton.edu can be used to access the WordNet.

use of this notion for word polarity detection. We explain these semantic relations in the following sections in detail.

2.2. Turkish WordNet

Turkish WordNet [14] is constructed within the BalkaNet [12] Project. BalkaNet project is basically constructed with the same approach as English WordNet and EuroWordNet project. Therefore, Turkish WordNet can be linked to BalkaNet WordNets, WordNets under EuroWordNet project, English WordNet, and any other WordNets linked to the ILI.

ILI is an excellent feature for integration of WordNets. By using this index, WordNets are connected to each other easily. With the help of this index, each synset in any WordNet is connected to the synset with similar meaning in any other WordNets. In this thesis, we take advantage of compatibility of WordNets by using ILI.

The comparison of the sizes of Turkish and English WordNets is illustrated in Table 2.2. It is seen that English WordNet is much more comprehensive than Turkish WordNet. In this thesis, to overcome the shortcomings of Turkish WordNet, we make use of the rich English WordNet for Turkish word polarity detection.

Table 2.2. Comparison of Turkish and English WordNet statistics.

Statistic	**Turkish**	**English**
Synset	14795	115424
Word-Sense pair	20420	203145

2.3. WordNet Relations

Table 2.3 shows a comparison of Turkish and English WordNets in terms of the number of semantic relations. We describe the semantic relations in Table 2.3 in more detail below. For each semantic relation, we provide an example for English and Turkish.

Table 2.3. Comparison of the semantic relations in Turkish and English WordNets.

Relations	**English**	**Turkish**
synonym	115424	14795
subevent	409	118
holo member	12205	989
holo part	8636	1497
holo portion	787	203
near antonym	7642	1227
verb group	1748	572
also see	3240	283
similar to	22196	84
hypernym	94842	11181
causes	218	96
be in state	1296	519
particle	106	0
eng derivative	36630	0
derived	6564	0
category domain	6166	357
usage domain	983	6
region domain	1280	0

- *synonym*: This relation implies a similar meaning between words.
  (i) English: car/automobile
  (ii) Turkish: araba/otomobil
- *subevent*: This relation implies the entailment relation between verbs. In this type of relation, one activity includes the other activity in it.
  (i) English: snore/sleep
  (ii) Turkish: horlamak/uyumak
- *holo member*: This relation implies membership relation between words.
  (i) English: America/NATO

(ii) Turkish: bitki/bitkiler alemi

- *holo part*: This relation shows a part-of relation.
  (i) English: finger/hand
  (ii) Turkish: kol/insan
- *holo portion*: This relation exists between wholes and their portions. If W1 has a W2 and W2 is a portion of W1, then this relation exists between W1 and W2.
  (i) English: cocain/coca
  (ii) Turkish: şarap/üzüm
- *near antonym*: This relation indicates opposite meanings between words.
  (i) English: sell/buy
  (ii) Turkish: artmak/azalmak
- *verb group*: This relation groups verbs according to the similarities of their meanings.
  (i) English: behave/act
  (ii) Turkish: nefes almak/solumak
- *also see*: This relation implies a semantic relation of related words.
  (i) English: cold/cool
  (ii) Turkish: hızlı/ani
- *similar to*: This relation implies semantic similarity between words. They have close referential meaning. Unlike also see relation, similar to relation exists between synsets that are not systemically related to other synsets through semantic relations.
  (i) English: rejected/disapproved
  (ii) Turkish: sıkıcı/monoton
- *hypernym*: This relation implies an IS-A relation between words.
  (i) English: berry/fruit
  (ii) Turkish: zurna/üflemeli çalgı
- *causes*: This relation exists between an event denoted by a word causing a resulting event.
  (i) English: kill/die
  (ii) Turkish: açıklamak/duyulmak

- *be in state*: This relation exists between words if they have "value of" relation.
  (i) English: tall/height
  (ii) Turkish: mutsuz/saadet
- *particle, eng derivative, derived*: These relations imply derivational relations between words.
  (i) English: emerging/emerge
  (ii) English: go/going
  (iii) English: quick/quickly
- *category domain, usage domain, region domain*: These relations imply relations about topical classifications of meanings.
  (i) English: diplomatic immunity/law
  (ii) Turkish: sanat eseri/sanat

In this thesis, we use synonym, hypernym, also see, similar to and derivation relations to construct a word relatedness graph by connecting similar words to each other. Our approach based on the fact that similar words tend to have similar polarities.

2.4. General Inquirer

General Inquirer [8] is a computer-assisted approach for content analyses. It is constructed by Philip Stone *et al.* and supported by grants from USA National Science Foundation. It contains 182 tag categories. Some of the important tag categories are summarized in Table 2.4.

When you submit a file including text to General Inquirer, it analyzes this document in terms of tagged categories and it reports a score for each tagged category.

In this thesis, we use General Inquirer for negative and positive tagged seed words. We obtain 1915 positive seed words and 2291 negative seed words from General Inquirer. A sample of seed words is shown in Table 2.5. We use these positive and negative seed words in a semi-supervised setting to predict the polarities of unlabeled words and for evaluation purposes. There is no resource such as General Inquirer for

Table 2.4. A sample set of General Inquirer categories.

Category	**Definition**
Positive	1915 words imply positive feelings
Negative	2291 words imply negative feelings
Active	2045 words imply activeness
Passive	911 words imply passiveness
Strong	1902 words imply strength
Weak	755 words imply weakness
Academy	153 words imply academic, intellectual or educational relations
Male	56 words imply men and roles associated with them
Female	43 words imply women and roles associated with them

Turkish. Therefore, we propose an approach to generate positive and negative seed words for Turkish as well as for other languages by making use of the English seed words and the compatibility of WordNets.

2.5. Types of Machine Learning Algorithms

Machine learning is a discipline that considers design and development of algorithms that allow computers to predict behaviors based on empirical data. It is used for several applications such as natural language processing and sentiment analysis. In Table 2.6, there is a comparison of three types of machine learning algorithms. Now we examine these algorithms in more detail.

- *Supervised learning*: It is a machine learning task that uses labeled training data. Each training data consists of an input and a corresponding output. A classifier is generated based on this information. With the help of this classifier, the algorithm predicts an output for each given valid input.
- *Unsupervised learning*: In contrast to supervised learning, unsupervised learning uses unlabeled training data. It tries to find hidden structures of the unlabeled training data.

Table 2.5. A sample set of seed words in General Inquirer.

Negative seeds	**Positive seeds**
annoy	beautiful
disappointment	comfort
disaster	congratulation
forbidden	courage
hell	empower
ill	fantastic
ineffective	funny
leak	like
madness	manageable
antisocial	opportunity
nervous	beneficial

Table 2.6. Types of machine learning algorithms.

Type	**Use of labeled data**	**Use of unlabeled data**
Supervised learning	✓	-
Unsupervised learning	-	✓
Semi-supervised learning	✓	✓

- *Semi-Supervised learning*: Semi-supervised learning uses both labeled and unlabeled data for training. Typically a small amount of labeled and huge amount of unlabeled data is used. It is an approach which resides between supervised and unsupervised learning.

In this thesis, we use a semi-supervised approach for word polarity detection. To find the polarity of a given word, we use both labeled seed words and unlabeled words in a word relatedness graph.

2.6. Cross-validation

Cross-validation, or rotation estimation, is an evaluation technique for estimating the performance of a predictive model. It is mainly used to estimate how accurately a model performs in practice. The main idea is to partition data set into subsets randomly and perform analysis on one subset called training set and validate the analysis on the other subset called validation set.

Some of the common cross-validation types are explained below:

- *K-fold cross-validation*: In k-fold cross validation, data set is randomly divided into k subsets. In each turn, one of the k subsets is selected as validation data and the other $k-1$ subsets are selected as training data. The cross-validation process is repeated k times in which each of the k subsets are used exactly once as validation data. Then,the k results can be averaged or combined for final estimation. The advantage of this method is that all subsets are used for both validation and training purposes and each subset is used as validation set exactly once. The most common version of k-fold cross validation is 10-fold cross validation.
- *Repeated random sub-sampling validation*: This approach divides the data set into training and validation sets. For each split, the model is trained using training data and validated by using validation data. Advantage of this method is that the proportion of training/validation set is not dependent to the number of folds. The disadvantage of this method is that in contrast to k-fold cross validation, some members of the data set may never be selected as validation data, whereas some of them may be selected more than once. This causes variation in the results when the analysis is repeated with different random splits.

In this thesis, we use 10-fold cross validation over the set of seed words to evaluate our approach. We choose 10-fold cross validation because all members of the data set are considered for validation exactly once. By using this approach, we evaluate all members of data set. The other reason why we choose 10-fold cross validation is to be able to compare our results with previous studies most of which use 10-fold cross

validation as well.

2.7. Random Walk Model

Markov chain is described as moving successively from one state to another in the system. The probability of moving from state i to state j is defined as transition probability denoted by p_{ij}. Each move in the system is called step. A famous example of Markov chain process is a frog jumping on the pads. One of the Markov chains, random walk [16], is defined as the sequence of vertices that are selected randomly in a graph traversal. When you start a random walk, you select a neighbor of the current vertex randomly and you move to this selected node. You iterate this process until a stopping condition is met. Random walk can be ended by several stopping conditions such as the length of random walk or reaching a specific node.

The pseudocode of random walk algorithm is illustrated in Figure 2.1.

```
Constitute G as the word relatedness graph;
Define i as the start vertex;
Define p as new Path;
while stopping condition is not met do
    Add i to the path p;
    Select i as a neighbor of the current i randomly on G;
end while
```

Figure 2.1. Random Walk algorithm.

3. RELATED WORK

One of the earliest works about word polarity detection was conducted by Hatzivassiloglou *et al.* [2]. They propose a method that finds semantic orientation of any adjective using information collected from a large corpus. They show that conjunctions of adjectives can be used to determine polarity of any given adjective. They focus on conjunctive expressions such as "simple and well-received" and "simplistic but well-received". The underlying assumption is that conjunctions of adjectives using "and" usually have similar semantic orientation, whereas adjectives conjoined using "but" usually have different semantic orientation. They achieve an accuracy of 82% for adjectives. Their algorithm to find polarity of any given adjective consists of the following stages.

- All conjunctions of adjectives are extracted from an English corpus.
- A graph is constructed with two types of links between adjectives which imply same or different orientation.
- A clustering is run on the graph to obtain two clusters of adjectives with different semantic orientation. Each cluster contains adjectives with same orientation and each cluster is connected to other via links of different orientations.
- Because its known that positive adjectives tend to be used more frequently than negative adjectives, the cluster with higher average frequencies is classified as positive and the other cluster is classified as negative.

Turney and Littman [1] propose an unsupervised algorithm to find semantic orientation of words. They define seven positive and seven negative paradigm seed words. These paradigm words are presented in Table 3.1. These words are chosen by considering all senses. These words have same orientation in all senses. For example, excellent has positive semantic orientation in almost all contexts. They find semantic orientation of any word by calculating its semantic association with a set of positive words minus its association to a set of negative words. If this value is positive, any

given word is classified as positive and if negative, any given words is classified as negative. Absolute value of this value can be considered as strength of its polarity. To calculate semantic association between any two words, they propose two approaches which are Pointwise Mutual Information and Latent Semantic Analysis. They achieve an accuracy of 82.8%.

Table 3.1. Set of paradigm words used by Turney and Littman [1].

Polarity	**Words**
Positive	good, nice, excellent, positive, fortunate, correct, superior
Negative	bad, nasty, poor, negative, unfortunate, wrong, inferior

In Pointwise Mutual Information approach, to calculate the strength of semantic association, Turney and Littman [1] calculate the strength of statistical dependence between two given words. To estimate statistical dependency between a given word and the paradigm words, they query the given word with paradigm words by using the near operator in the Altavista search engine. Altavista search engine is chosen because it has a near operator. The near operator finds documents on the web such that each document contains the given words within ten words of one another, in either order. They count the number of hits by using the near operator with the given word and paradigm words and they also count the hits of the given words alone. Using these two values, they calculate statistical dependence between the given word and paradigm words. If the given word tends to co-occur with positive paradigm words, it is classified as positive and it is classified as negative otherwise. In Latent Semantic Analysis approach, they analyze statistical relationships among a given word with paradigm words in a corpus by using Singular Value Decomposition and according to the results, they classify the given word as positive or negative.

Takamura *et al.* [9] propose a method which regards semantic orientation as spin of electrons, where neighbor electrons tend to have the same spin. They consider each word as an electron and its polarity as a spin value and classify a word as positive or negative with the help of its spin value. They use labeled seed words as a beginning

point for the spin model. They calculate the spin value using mean-field approximation with iterative update rule. They construct three networks by linking words with the help of glosses and relations in WordNet [7]. They also enhance the networks by using conjunctive information extracted from a corpus. They have two types of links which are same semantic orientation links and different orientation links. The network structures are explained in the following items.

- The Gloss network is constructed using gloss definitions. If a word appears in the gloss of any other word, these two words are connected. If a word preceded a negation word, type of the edge is selected as different orientation link and others are selected as same orientation links.
- The Gloss-Thesaurus network extends the Gloss Network structure by including synonym, antonym and hypernym relations from WordNet. Synonym and hypernym relations are treated as same orientation links whereas antonym relations are treated as different orientation links.
- The Gloss-Thesaurus-Corpus network extends the Gloss-Thesaurus network by including connections extracted from the conjunctive expressions in a corpus. To be able to do this, they follow Hatzivassilogou *et al.*'s method [2]. If two adjectives are connected by "and", the edge between adjectives is constructed as same orientation link whereas if they are connected by "but", the edge between adjectives is established as different orientation link.

To calculate spin value, a labeled data set is required. Takamura *et al.* [9] use negative and positive seed words in General Inquirer [8] for both learning and evaluation purposes. Initially the spin value of positive seed words is set to 1 and the spin value of negative seed words is set to -1. Spin value of other words are set to 0. They iteratively update the spin value of all words and when the difference between any iterative spin values is less than a threshold, they regard the computation as converged. Then, the words with high final average spin values are regarded as positive words and the ones with low final average spin values are regarded as negative words. They evaluate their algorithm using various number of seed words. They also use 10-fold cross validation

for evaluation. They achieve an accuracy of 91.5%.

Kamps *et al.* construct a word relatedness graph by using WordNet synonym relations to identify the semantic orientations of adjectives [17]. They use minimum shortest path as a proximity measure. A given word is classified as positive, if its minimum distance to the adjective "good" is less than its minimum distance to the adjective "bad". In detail, they obtain difference of distances for any given word by subtracting minimum distance to adjective "bad" from minimum distance to adjective "good". They obtain a polarity score by dividing this difference to minimum distance between reference words "bad" and "good". If this value is between 0 and 1, it is classified as positive and if it is between 0 and -1, it is classified as negative. They evaluate their approach by using positive and negative tagged words in General Inquirer [8]. They achieve an accuracy of 68.19% for adjectives.

Hassan and Radev [10] propose a semi-supervised method where random walk model is used to find polarity of any given word. They use mean hitting time to estimate word polarity. They construct a word relatedness graph by using relations in WordNet [7]. They also use General Inquirer [8] to obtain labeled seed words and to evaluate their method. Hitting time between two nodes is defined as average number of steps that a random walk takes from originator node to target node for the first time. They calculate mean hitting times from a given word to set of negative and positive seeds. According to the mean hitting times, they classify a given word as positive or negative. To calculate mean hitting time, they use a Monte Carlo based sampling algorithm for estimating it. They make several experiments to evaluate performance of their method. They compare their method with Pointwise Mutual Information model [1], spin model [9] and shortest path model [17]. They show that random walk with hitting time performs better than the methods reported previously in the literature. Therefore, in this thesis, we compare our approach with the hitting time based method. They achieve an accuracy of 93.1%.

Hassan *et al.* [18] propose an algorithm to find semantic orientation of Non-English words and evaluate their approach for Arabic and Hindi with a set of 300

manually labeled seed words for each language. They use random walk model with hitting time for polarity detection. They construct a multilingual network by connecting English and Non-English words by using a dictionary. For every Non-English word, they look up its possible meanings in the dictionary and connect this Non-English word to its possible meanings.

Our multilingual approach is different than most of the previous studies in three aspects: First, we develop a new approach to establish connections between Non-English and English words. We propose to use ILI for multilingual connections. With the help of this index, WordNets are easily and effectively connected to each other by linking the words in one WordNet to their similar meanings in the other WordNets. Second, we use Turkish as a Non-English language and generate a list of 2812 semi-automatically labeled seed words. Third, we propose using commute time as a proximity measure with random walk model for word polarity detection.

4. APPROACH

4.1. Overview

In this work, we construct a word relatedness graph for word polarity detection. We connect words if they are semantically related. The reason why we connect similar words is that similar words tend to have similar polarity. We use WordNet as a source of relations for connecting words. We extend the graph by connecting the WordNets of different languages with the help of the ILI based on English WordNet. We develop a semi-automated procedure to produce a set of positive and negative seed words for Non-English languages by using a set of English seed words. We use random walk model with truncated commute time as a proximity measure for word polarity detection. For any given word, we start a random walk from it. We stop the random walk when we return to the given word and the path contains a labeled word. If the labeled node in the path is positive, the length of this path is used to estimate the commute time from the given word to the positive set of seeds. Similarly, if the labeled node in the path is negative, it is used to estimate commute time from the given word to the negative set of seeds. In the end, according to average estimation of commute time, the given word is classified as positive if the commute time to the positive set of seeds is smaller than the commute time to the negative set of seeds and it is classified as negative otherwise. We use cross validation for our experiments. We examine our approach in detail in the following sections.

4.2. Monolingual Graph Construction

We construct an undirected weighted graph $G = (V, E)$ comprising a set V of vertices and a set E of edges. Vertices correspond to word and part-of-speech pairs in WordNet. Two words are connected if they have one or more of the synonym, hypernym, also see, similar to and derivation relations in WordNet. Weight of an edge between two words is directly proportional to the number of WordNet relations between them. A sample portions of the Turkish and English monolingual graphs are

presented in Figure 4.1.

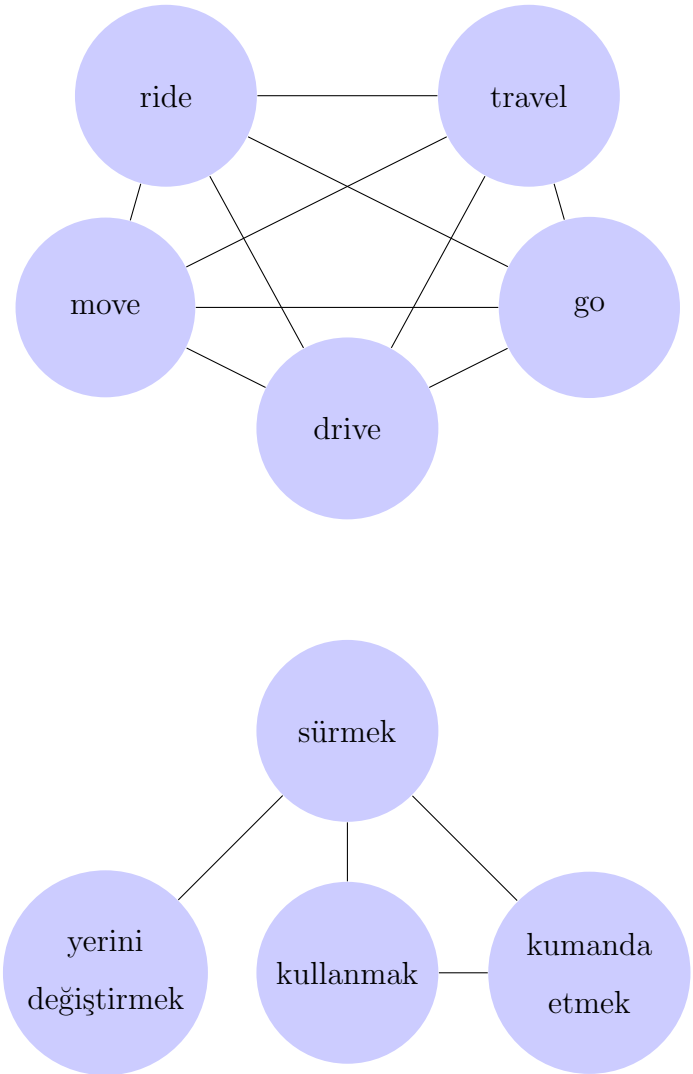


Figure 4.1. A sample view of English and Turkish word relatedness graphs.

4.3. Multilingual Graph Construction

Non-English WordNets are in general not as comprehensive as the English WordNet. However, most WordNets such as EuroWordNet [19] and BalkaNet [12] are designed to be compatible with English WordNet. This compatibility provides a simple and effective way to integrate such WordNets to the powerful English WordNet. As seen in Figure 4.2, we extend our word relatedness graph by connecting the words in English WordNet with similar words in Non-English WordNet by using the ILI. With the help of this index, it is possible to reach from a synset in any WordNet

to the synsets of the same meaning in the other WordNets. The multilingual graph construction algorithm is outlined in Figure 4.3.

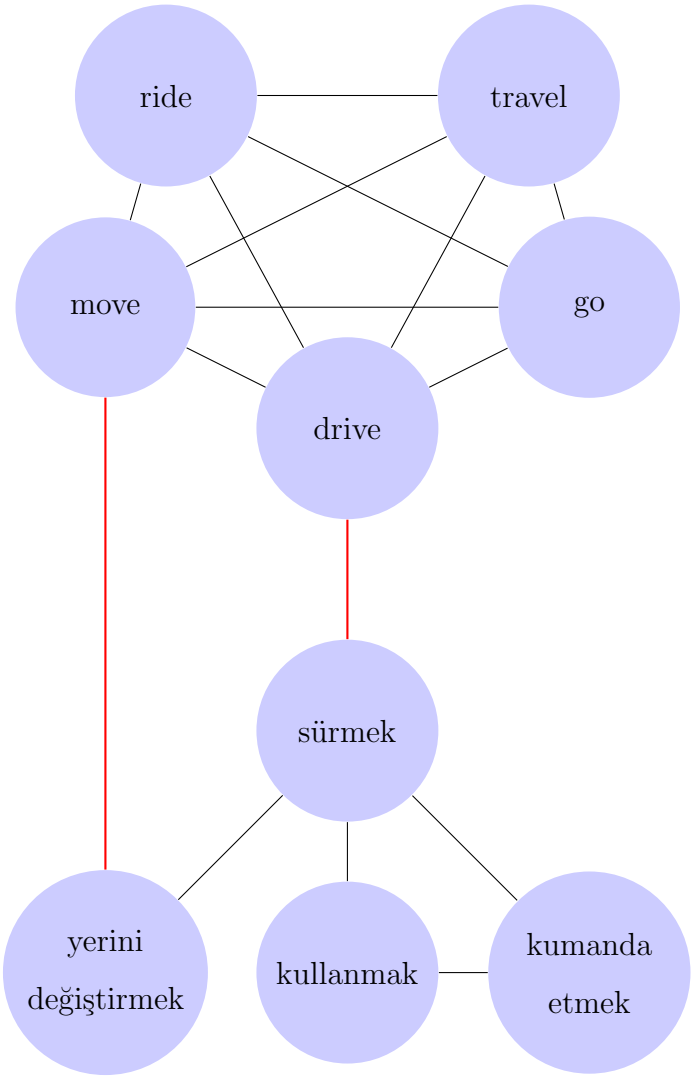


Figure 4.2. A sample view of the multilingual word relatedness graph with monolingual (black) and multilingual (red) links.

4.4. Commute Time Model

Consider a random walk [16] on graph G. If we are on vertex i, the probability of moving to the neighbor vertex j in the next step is directly proportional to the weight of the edge between i and j. Thus, the transition probability p_{ij} of moving from vertex i to vertex j is as follows:

```
Constitute engWN as English WordNet;
Constitute nengWN as Non-English WordNet;
for all Synset i ∈ nengWN do
    Synset j ⇐ findSynset(ILI of i, engWN);
    for all Word w1 ∈ i do
        for all Word w2 ∈ j do
            Connect w1 to w2 on monolingual graphs;
        end for
    end for
end for
procedure findSynset(ILI, wordNet)
    return synset by using ILI and wordNet;
end procedure
```

Figure 4.3. Multilingual graph construction algorithm.

$$p_{ij} = \frac{W_{ij}}{\sum_k W_{ik}} \tag{4.1}$$

where W_{ij} is the weight of edge between vertices i and j, and k denotes all the neighbors of vertex i. Now we explain two proximity measures originating from random walks, hitting time and commute time.

Hitting time between vertex i and vertex j, denoted by h_{ij}, is the expected number of steps in a random walk before vertex j is visited for the first time starting from vertex i [20, 21]. It can be calculated recursively as:

$$h_{ij} = \begin{cases} 0 & \text{if } i = j \\ 1 + \sum_k p_{ik} h_{kj} & \text{if } i \neq j \end{cases} \tag{4.2}$$

where k denotes all neighbors of vertex i. Hitting time has been used to find word polarity by Hassan and Radev [10], who have shown that it achieves the state of the art performance in the literature. A drawback of hitting time is that it is not symmetric. It is possible to end up with situations where vertex i is close to vertex j (h_{ij} is small), but vertex j is far away from vertex i (h_{ji} is big). We propose using the commute time proximity measure, which is a symmetric extension of hitting time.

Commute time between vertex i and vertex j, denoted by c_{ij}, is the expected number of steps in a random walk to reach vertex j for the first time starting from vertex i and return to vertex i again [20, 21]. It can be calculated by using hitting time:

$$c_{ij} = h_{ij} + h_{ji} \tag{4.3}$$

As illustrated in Equation 4.3, commute time is symmetric version of hitting time. In other words, c_{ij} is equal to c_{ji} for any vertex i and j in a graph. We use commute time as a proximity measure because it defines similarity between vertices i and j by considering a two dimensional perspective. One of the dimensions is the path from vertex i to vertex j and the other one is the path from vertex j to vertex i.

Hitting and commute time have a disadvantage. They are sensitive to long paths far away from the starting node [20, 21]. In general, similar words tend to be close to each other on a word relatedness graph. This leads us to T-truncated hitting and commute time which only consider paths shorter than T. Therefore, hitting time and

commute time definitions are changed accordingly:

$$h_{ij}^{T} = \begin{cases} 0 & \text{if } i = j \text{ and T=0} \\ 1 + \sum_k p_{ik} h_{kj}^{T-1} & \text{if } i \neq j \end{cases} \tag{4.4}$$

$$c_{ij}^{T} = h_{ij}^{T} + h_{ji}^{T} \tag{4.5}$$

It is seen that if distance between vertex i and j is more than T hops away, then hitting time is approximated as T.

4.5. Algorithm

To find the polarity of a given word, we start a random walk from that word and compute the commute time to the set of positive (P) and negative (N) seed words. Let $c_{i|P}$ be the average of truncated commute times from i to each seed in P and $c_{i|N}$ be the average of truncated commute times from i to each seed in N. If $c_{i|P}$ is less than $c_{i|N}$ word i is classified as positive, otherwise it is classified as negative. The main idea is that commute time between two words shows how similar they are to each other. When the graph and the size of the seed list is large calculation of $c_{i|P}$ and $c_{i|N}$ is time consuming. As discussed in the next section, we use a sampling approach to estimate $c_{i|P}$ and $c_{i|N}$.

4.6. Sampling Approach

We propose a sampling approach to estimate the truncated commute time from any given word to set of seed words similar to previous works [10, 20]. We start M independent random walks with maximum length of T. Hitting one of the labeled seed words and returning to the starting word is the stopping condition. The length of a

random walk in which the stopping condition is not met is estimated as T.

Let's assume that m number of M random walks met the stopping condition and the length of each random walk is $\langle t_1, t_2, ..., t_{m-1}, t_m \rangle$. S denotes the set of positive and negative seed words. Then truncated commute is estimated as:

$$c^*_{i|S} = \frac{\sum_{i=1}^{m} t_i}{M} + (1 - \frac{m}{M})T \tag{4.6}$$

Our algorithm for word polarity detection is outlined in Figure 4.4. We start M random walks with maximum length of T on G to calculate the commute time from given word i to set of positive seed words $c^*_{i|P}$. Then same process is repeated once more to calculate the commute time from given word i to the set of negative seed words $c^*_{i|N}$. It identifies word polarity by comparing $c^*_{i|P}$ and $c^*_{i|N}$. If $c^*_{i|N}$ is less than $c^*_{i|P}$ word i is classified as negative, it is classified as positive otherwise. In the next chapter, we evaluate our word polarity detection algorithm by using various parameters.

Define **i** as the given word;
Calculate $\mathbf{c^*_{i|P}}$ using Eq. 4.6;
Calculate $\mathbf{c^*_{i|N}}$ using Eq. 4.6;
if $\mathbf{c^*_{i|P} > c^*_{i|N}}$ **then**
    Classify **i** as negative;
else
    Classify **i** as positive;
end if

Figure 4.4. Polarity detection using random walk with estimated commute time.

5. EXPERIMENTS

We apply our approach to detect polarities of English and Turkish words. We use the WordNets of each language, English WordNet [7] and Turkish WordNet [14], to construct monolingual word-relatedness graphs. We use synonym, hypernym, also see, similar to and derivation relations in each WordNet. The weight of an edge connecting two words is directly proportional to the number of relations between them.

A multilingual graph is obtained by connecting the monolingual graphs with ILI. We use General Inquirer as a source for English seed words. Like in previous works [1, 10], we ignore some ambiguous words and end up with 2085 negative and 1730 positive words. Like most Non-English languages, Turkish does not have a resource such as General Inquirer to obtain seed words. Figure 5.1 summarizes the semi-automated method that we propose to produce Non-English seed words using the ILI. By using this algorithm, we generate 1398 positive and 1414 negative seed words for Turkish. A sample of Turkish seed words is illustrated in Table 5.1. We use random walk model over the monolingual graphs and the English-Turkish multilingual graph to identify the polarities of words. We propose using commute time as a proximity measure and compare it with hitting time that was shown to outperform the previous approaches for English word polarity detection in [10]. We use 10-fold cross validation in our experiments and report the accuracies of polarity detection for the English and Turkish seed words both when the monolingual and the multilingual graphs are used.

5.1. Monolingual Experiments

Firstly, we compare our commute time method with the state of the art hitting time method [10] in the monolingual setting. Our algorithm and the hitting time algorithm depend on two parameters, number of random walks M and maximum length of a random walk T. We compare hitting time and commute time for various combinations of M and T.

```
Constitute engWN as English WordNet;
Constitute nengWN as Non-English WordNet;
Define P as the list of positive English seed words;
Define N as the list of negative English seed words;
positiveSeeds ⇐ generateNonEnglishSeeds(engWN, P, nengWN);
negativeSeeds ⇐ generateNonEnglishSeeds(engWN, N, nengWN);
Process positiveSeeds and negativeSeeds manually;
procedure generateNonEnglishSeeds(engWN, S, nengWN)
    Define seedList as the list of seed words;
    for all Word i ∈ S do
        SynsetList synsetList ⇐ findSynsetList(i, engWN);
        for all Synset k ∈ synsetList do
            Synset j ⇐ findSynset(ILI of k, nengWN);
            for all Word z ∈ j do
                Add z to the seedList;
            end for
        end for
    end for
    return seedList;
end procedure
procedure findSynsetList(i, wordNet)
    Define synsetList as Synset list;
    for all Synset s ∈ wordNet do
        if s contains i then
            Add synset s to synsetList;
        end if
    end for
    return synsetList;
end procedure
```

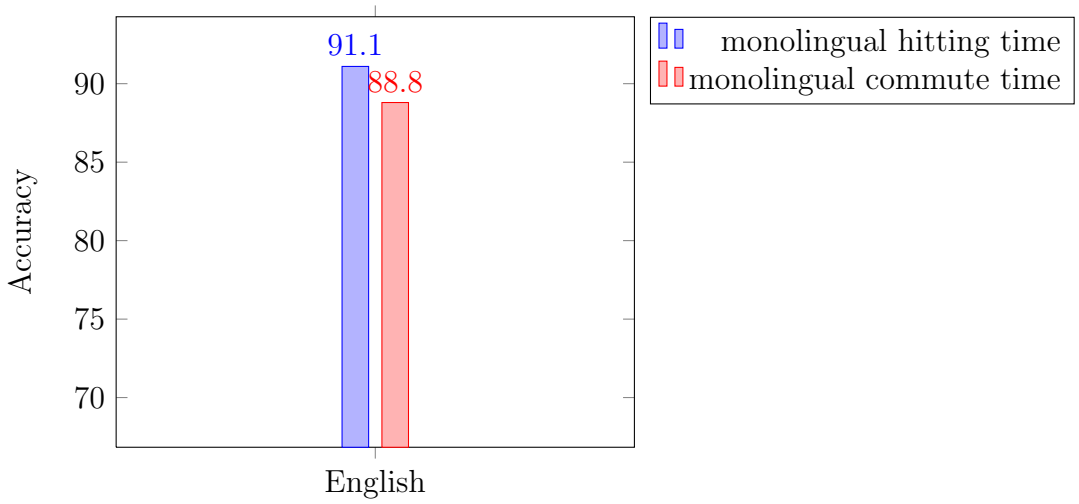
Figure 5.1. Seed generation algorithm.

Table 5.1. A sample set of Turkish seed words.

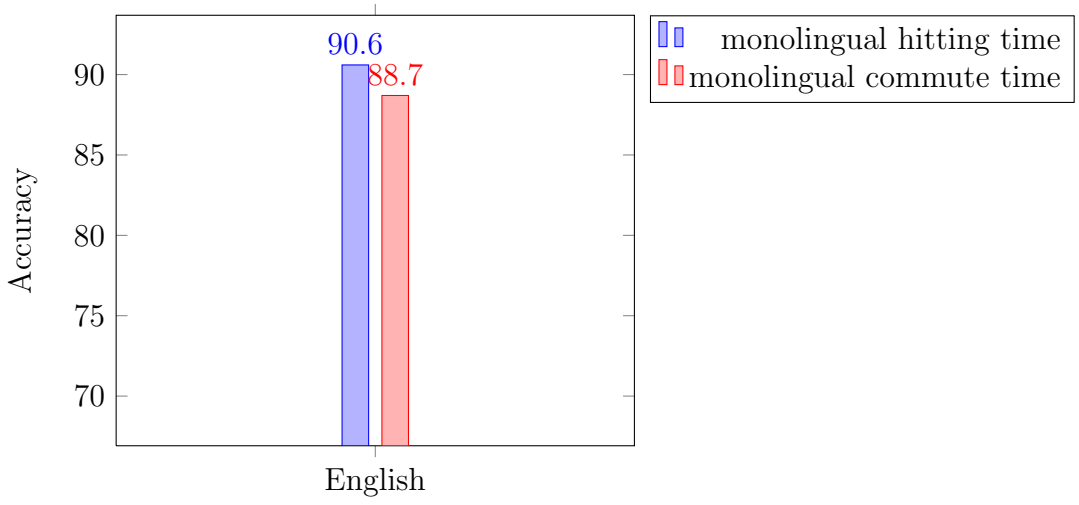
Negative seeds	**Positive seeds**
dalgın	kibar
gülünç	öpücük
aptalca	buse
saçma	öpme
anlamsız	öpüş
saçmalık	biliş
aptallık	vukuf
ahmaklık	gülüş
hakaret	gülmek
eziyet	yasal
kötü davranma	kanuni
suiistimal etmek	legal

Our experimental results are summarized in Figure 5.2, 5.3, 5.4 and 5.5. The proposed commute time algorithm performs similarly to the hitting time method. The maximum accuracy for English when the monolingual graph is used is 89.7%, which is comparable to 91.1% achieved by hitting time[3]. The maximum accuracy for Turkish when the monolingual graph is used is 86.6%, which is slightly better than 84.5% achieved by hitting time. In contrast to English, performance of commute time is better than hitting time for Turkish, because the size of the Turkish word relatedness graph and the average node degree are smaller than those in English. This increases the probability of finding paths that met the commute time stopping condition for Turkish. Turkish WordNet is not as rich as English WordNet. Therefore, the accuracies for Turkish are lower than the ones for English when we use the monolingual graphs.

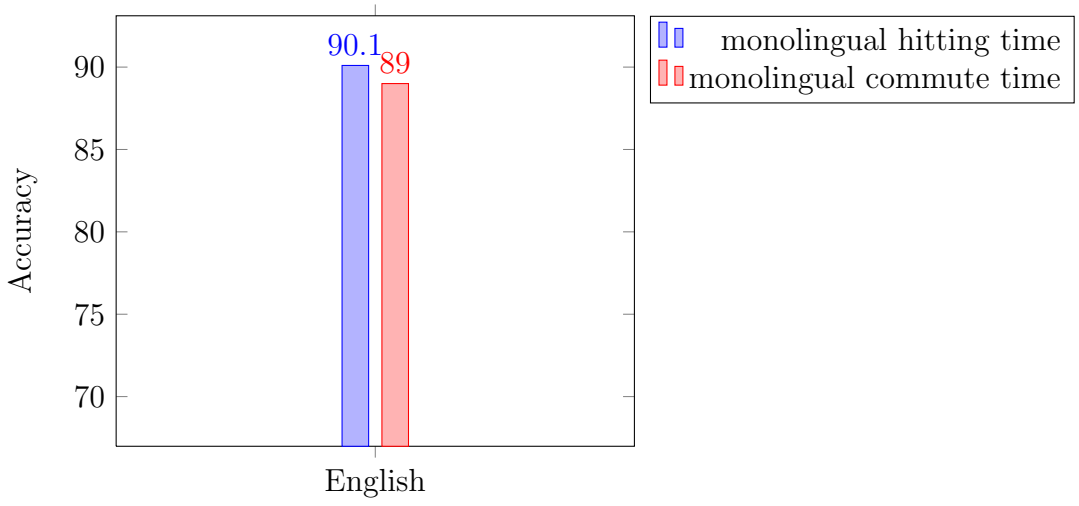
[3]The accuracy for English when hitting time is used is reported as 93.1% in (Hassan and Radev, 2010). The difference might be due to a different version of WordNet or the seed list.



(a) T=15 and M=1000

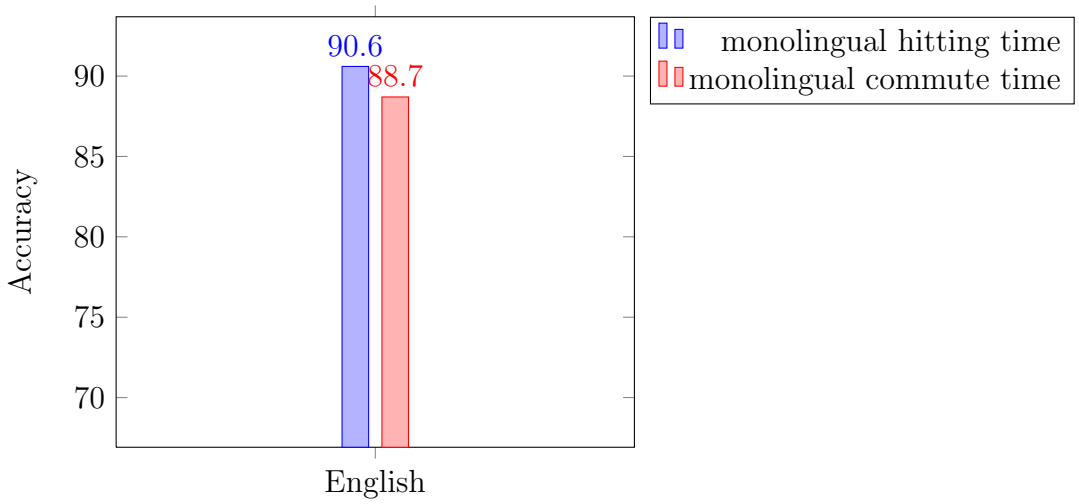


(b) T=30 and M=1000

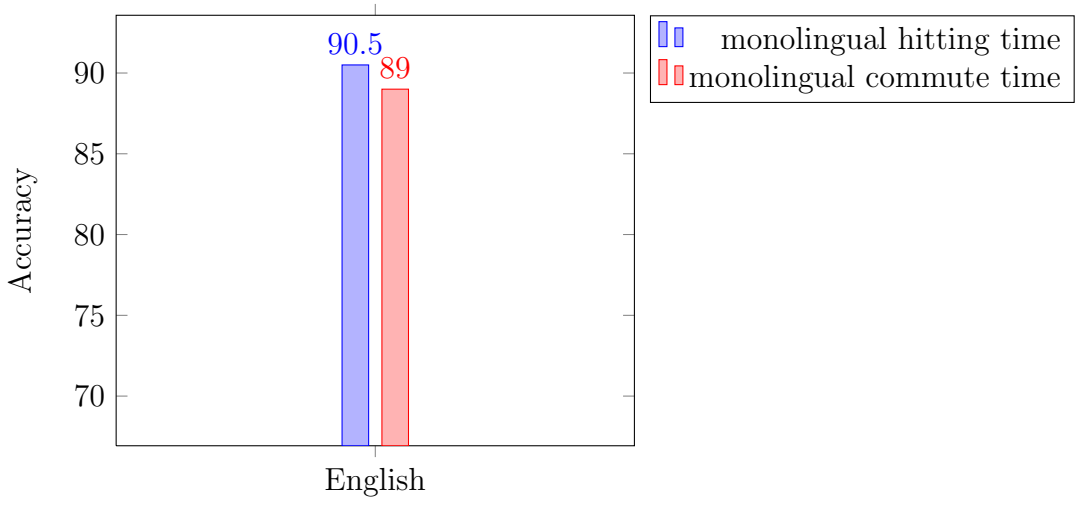


(c) T=45 and M=1000

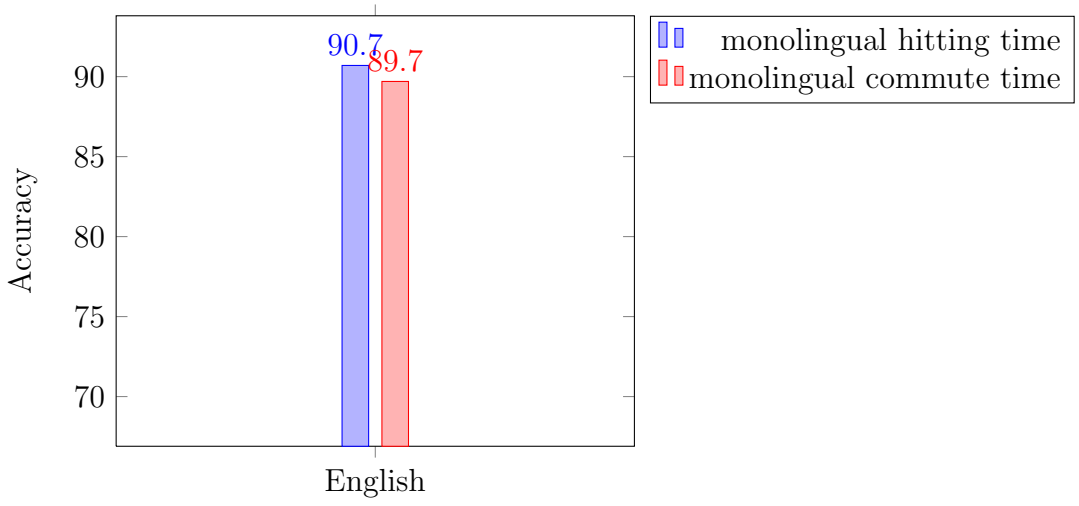
Figure 5.2. Monolingual accuracies using M=1000 and varying T for English.



(a) T=30 and M=1000

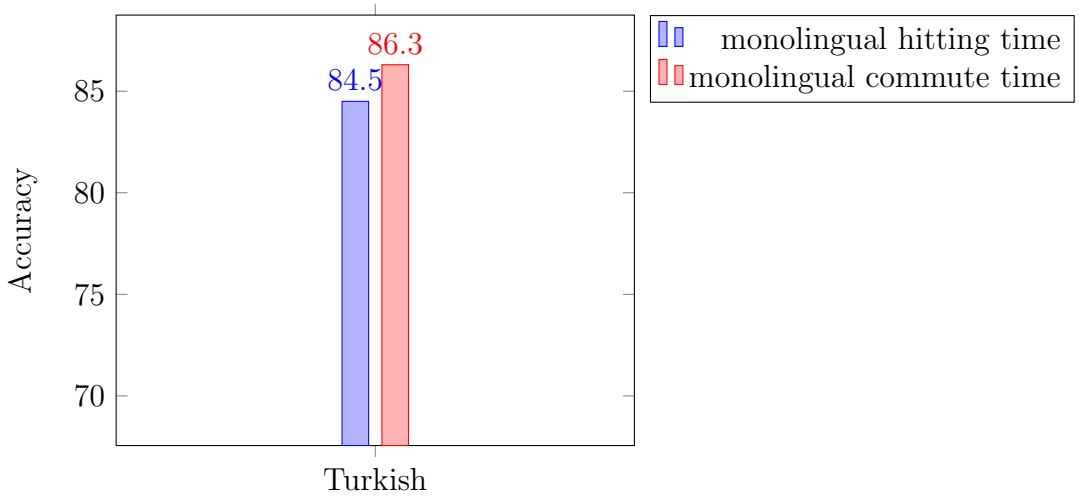


(b) T=30 and M=2000

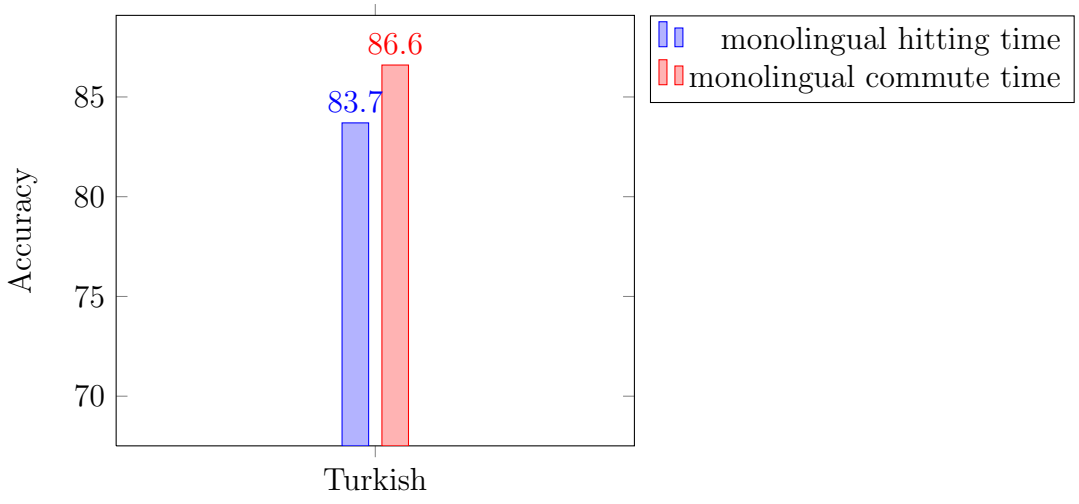


(c) T=30 and M=3000

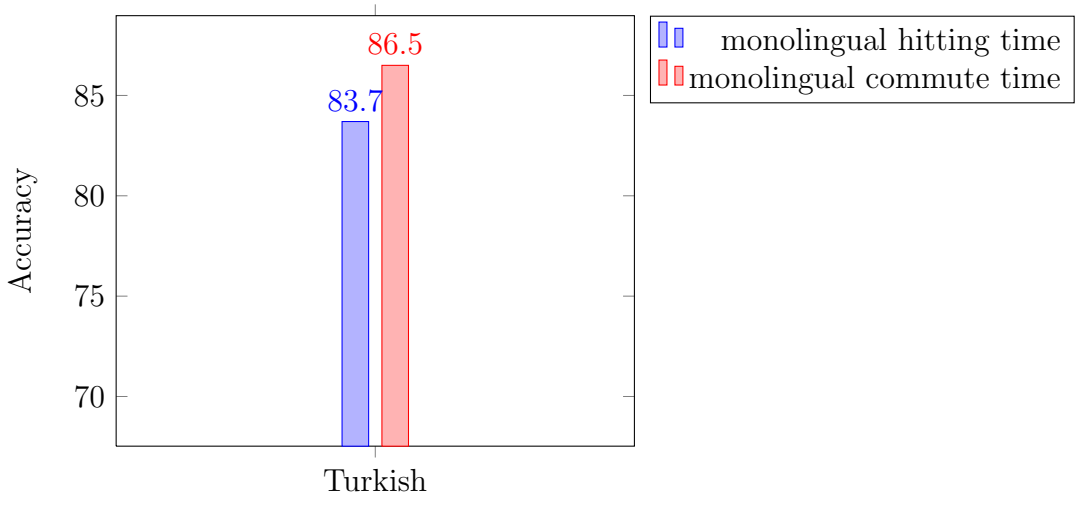
Figure 5.3. Monolingual accuracies using T=30 and varying M for English.



(a) T=15 and M=1000

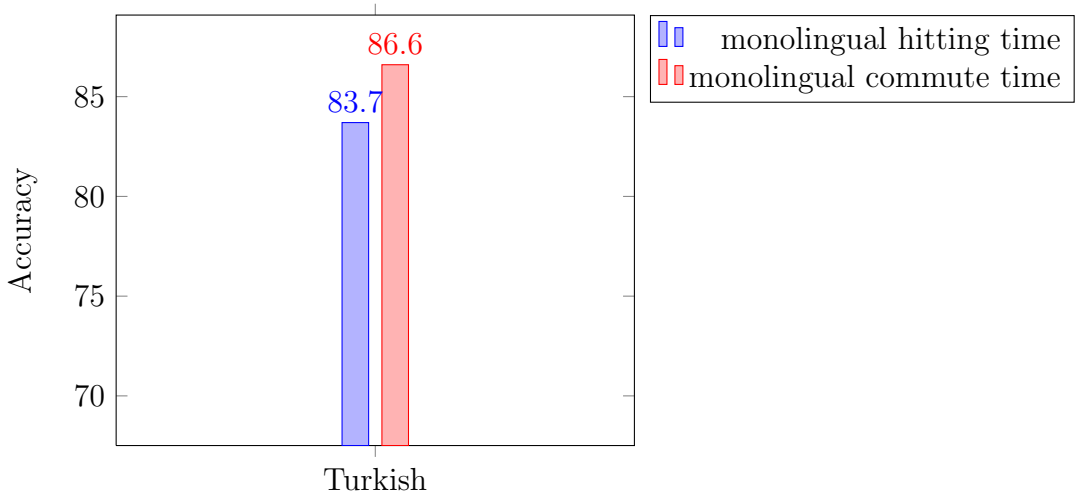


(b) T=30 and M=1000

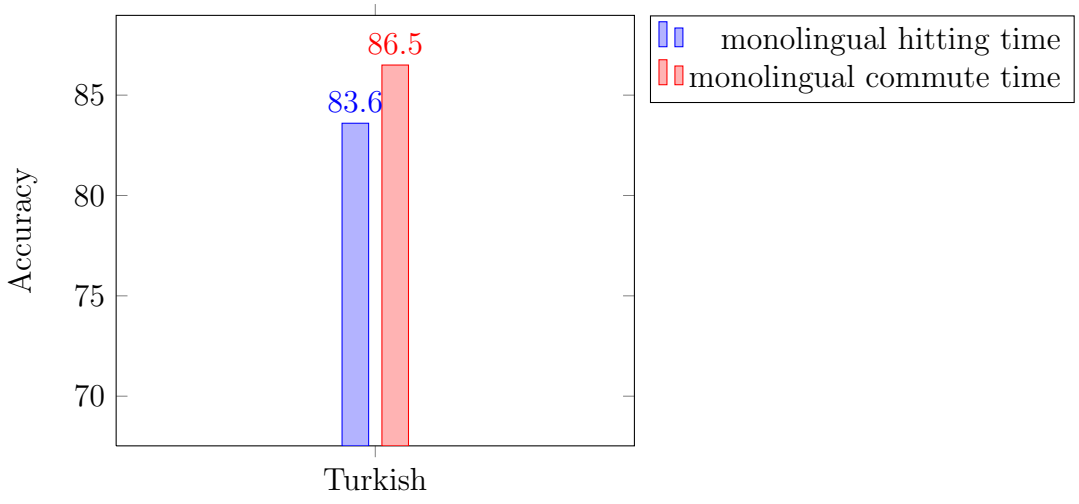


(c) T=45 and M=1000

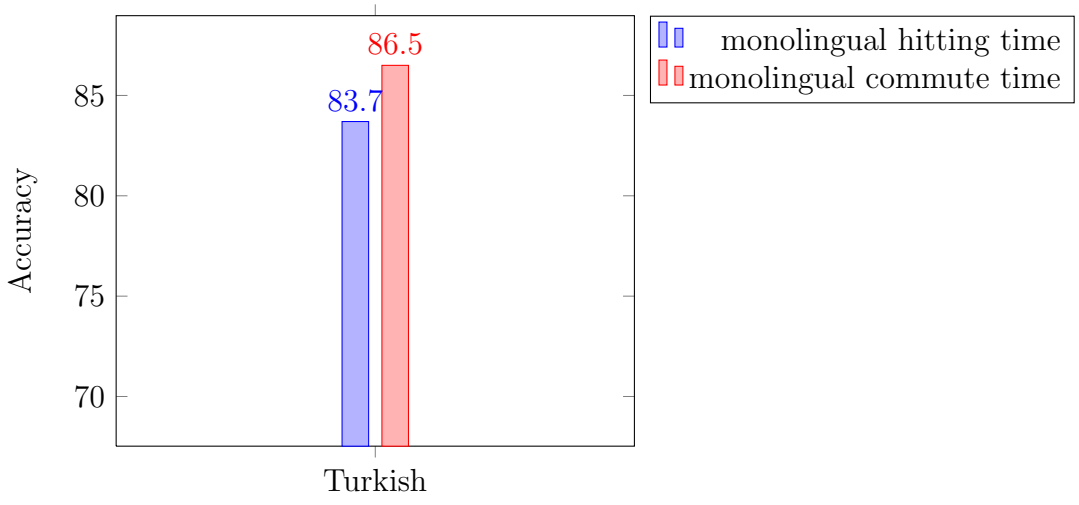
Figure 5.4. Monolingual accuracies using M=1000 and varying T for Turkish.



(a) T=30 and M=1000



(b) T=30 and M=2000



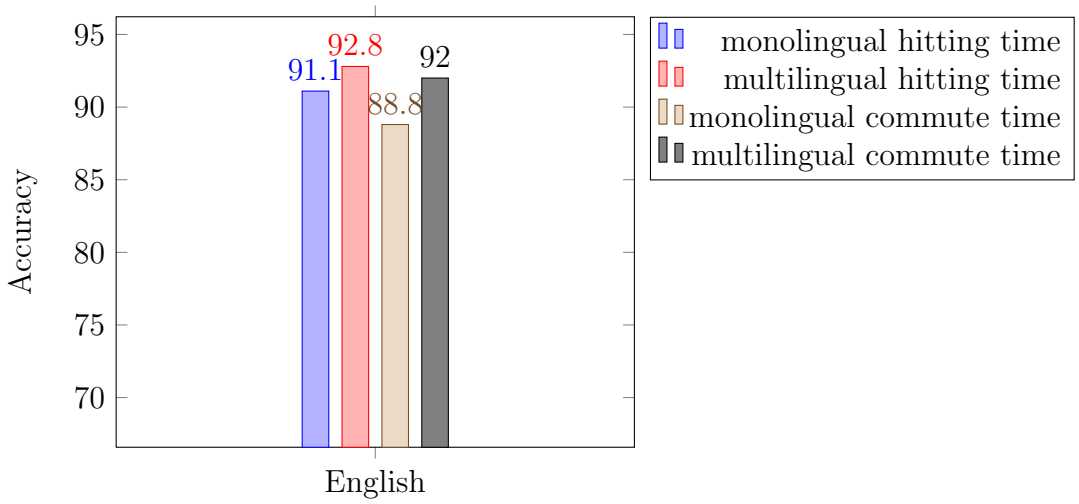
(c) T=30 and M=3000

Figure 5.5. Monolingual accuracies using T=30 and varying M for Turkish.

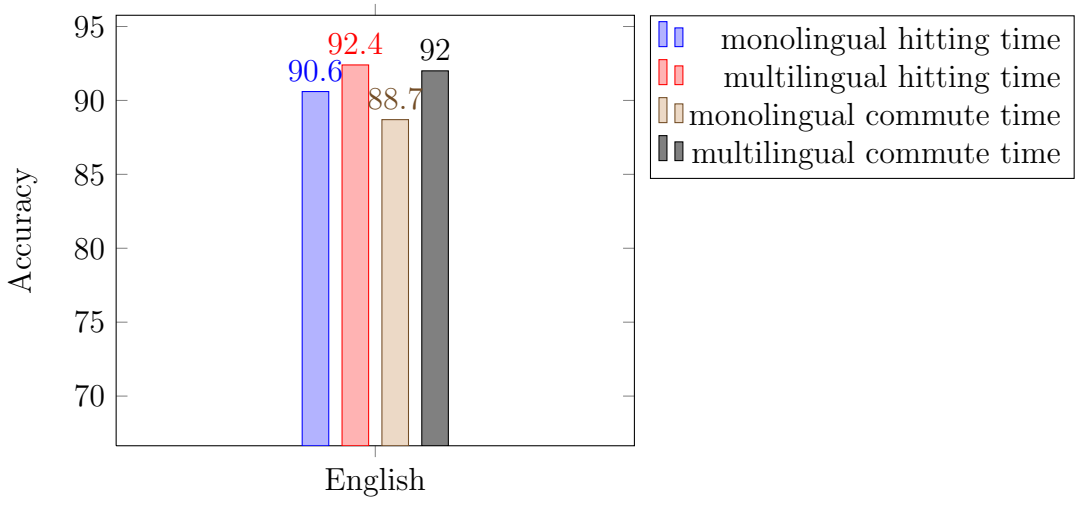
5.2. Multilingual Experiments

Non-English languages often do not have enough resources for linguistic studies. For example, the contents of Non-English WordNets are in general not as rich as the content of the English WordNet. We believe that integrating Non-English WordNets with the powerful English WordNet can significantly improve the performances of algorithms based on linguistic resources. As explained before, we obtain a multilingual graph by integrating Turkish and English WordNets with the help of ILI. By using ILI, we can access same synsets across Turkish and English WordNets. We evaluate commute time and hitting time methods using the constructed multilingual graph for various M and T values.

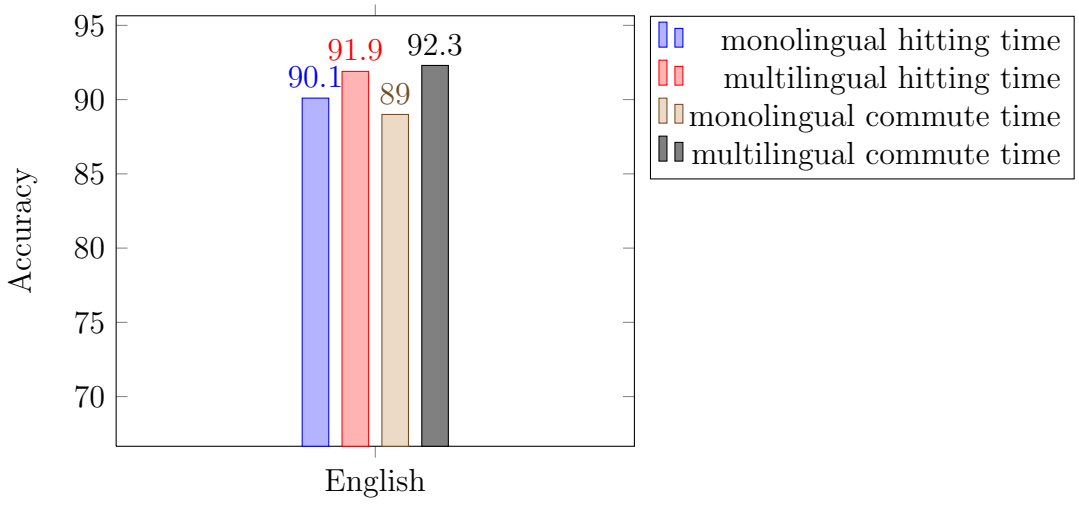
Our experimental studies in Figure 5.6, 5.7, 5.8 and 5.9 show that the multilingual approach leads to improvements for both languages. The improvement for Turkish is more significant since we take advantage of the dense English graph. Maximum accuracy for Turkish is improved from 86.6% to 95% with the multilingual commute time method, and it is improved from 84.5% to 95.5% with the multilingual hitting time method. Maximum accuracy for English is improved from 89.7% to 92.3% with the multilingual commute time method, and from 91.1% to 92.8% with the multilingual hitting time method. These results demonstrate that the richness of the English WordNet is a valuable resource for Turkish word polarity detection. Interestingly, Turkish WordNet is also able to boost the performance for English word polarity detection.



(a) T=15 and M=1000

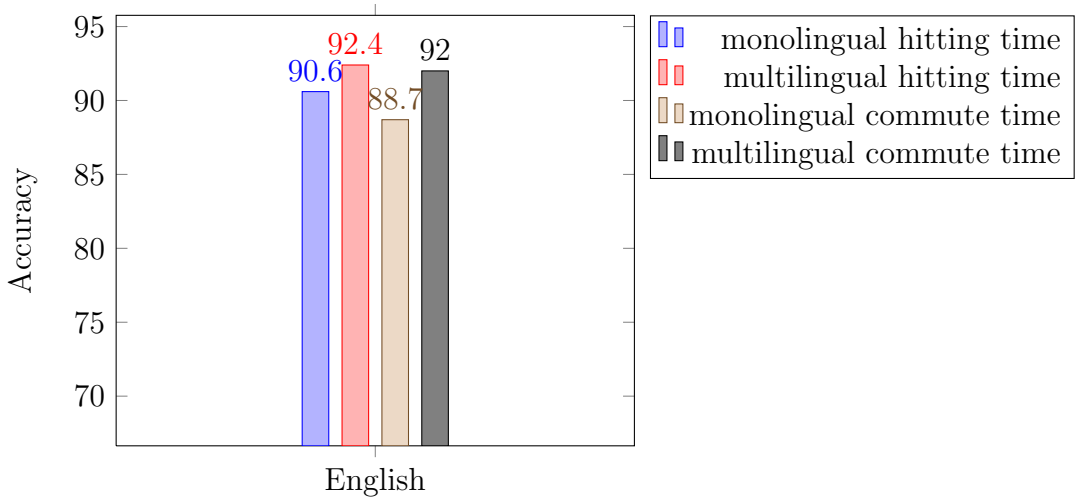


(b) T=30 and M=1000

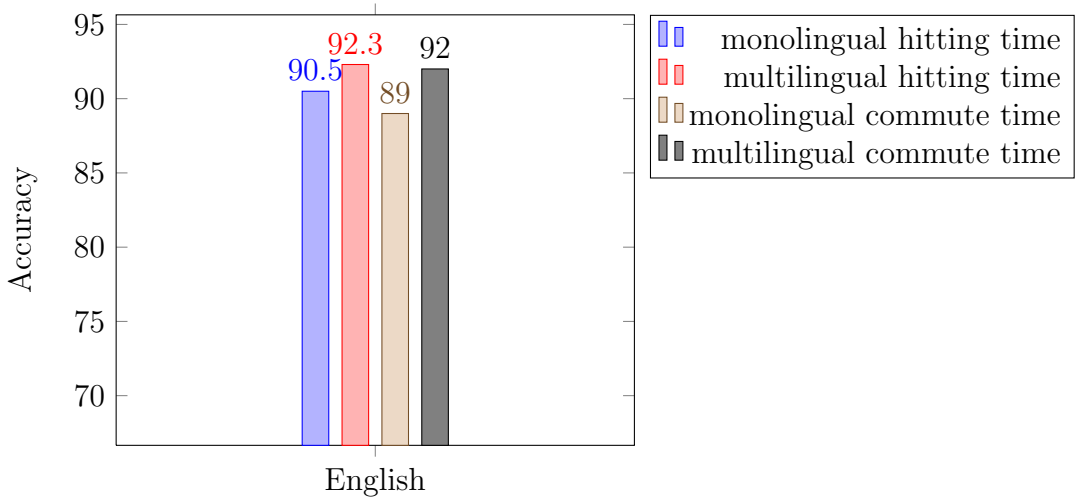


(c) T=45 and M=1000

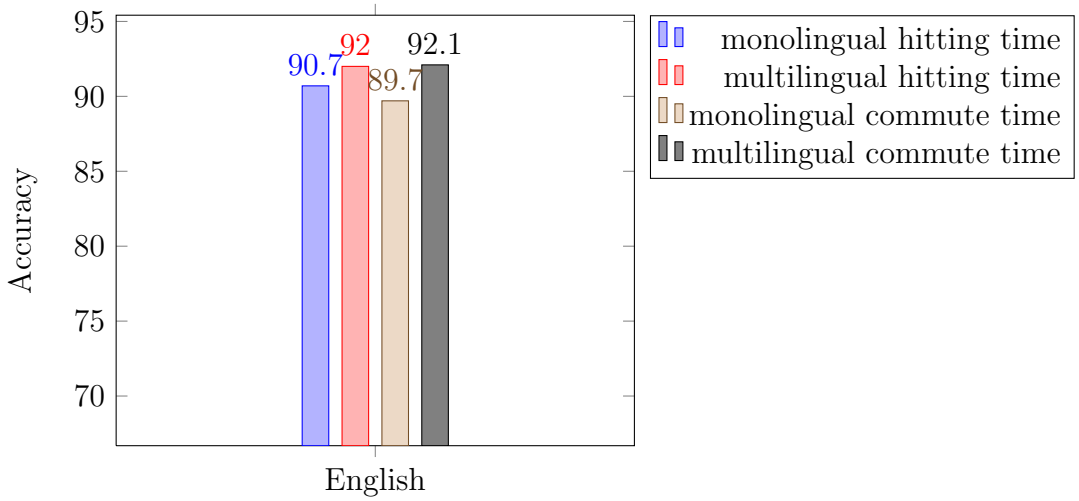
Figure 5.6. Multilingual accuracies using M=1000 and varying T for English.



(a) T=30 and M=1000

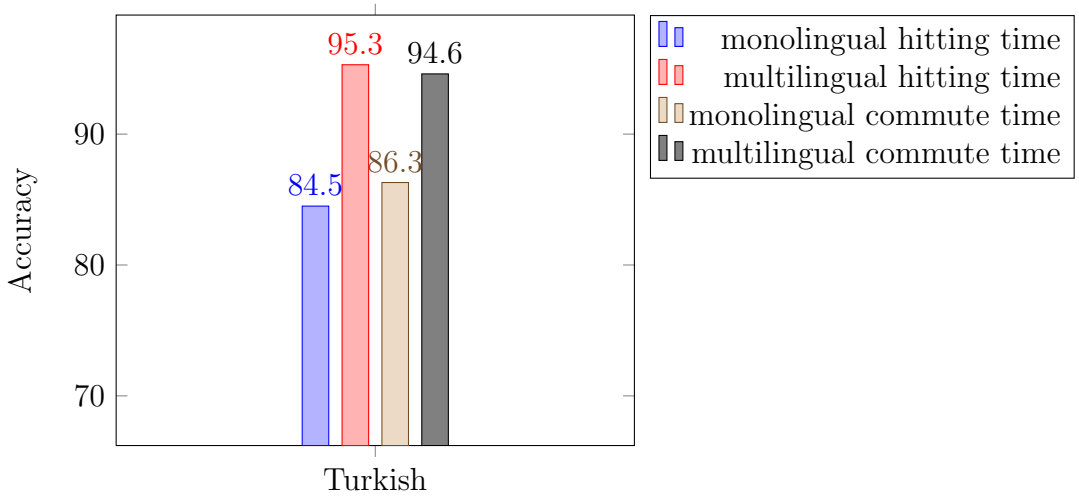


(b) T=30 and M=2000

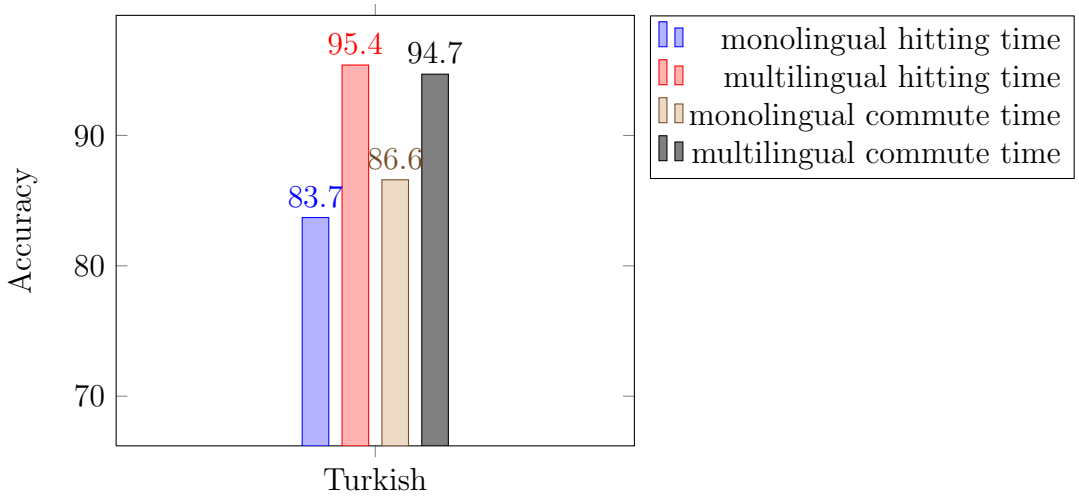


(c) T=30 and M=3000

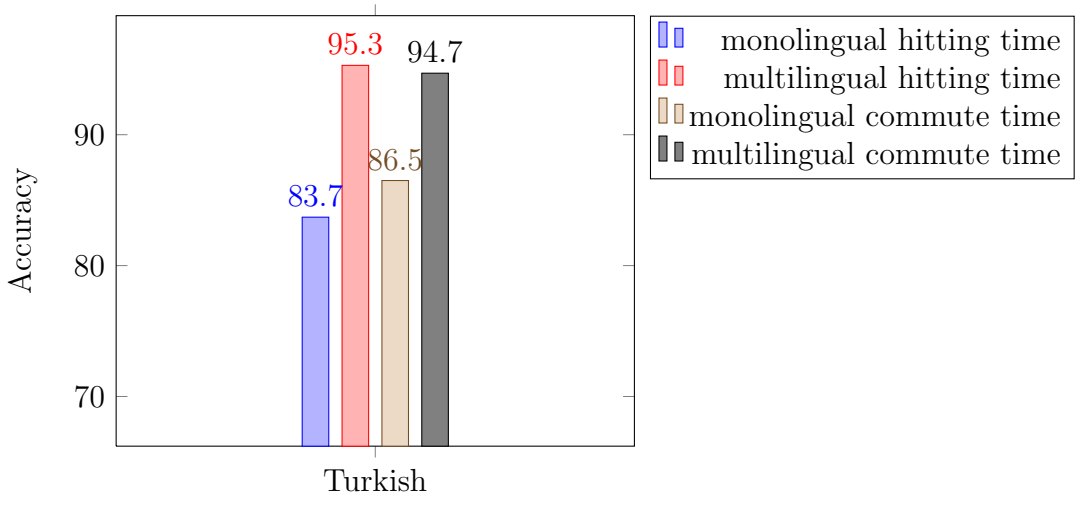
Figure 5.7. Multilingual accuracies using T=30 and varying M for English.



(a) T=15 and M=1000

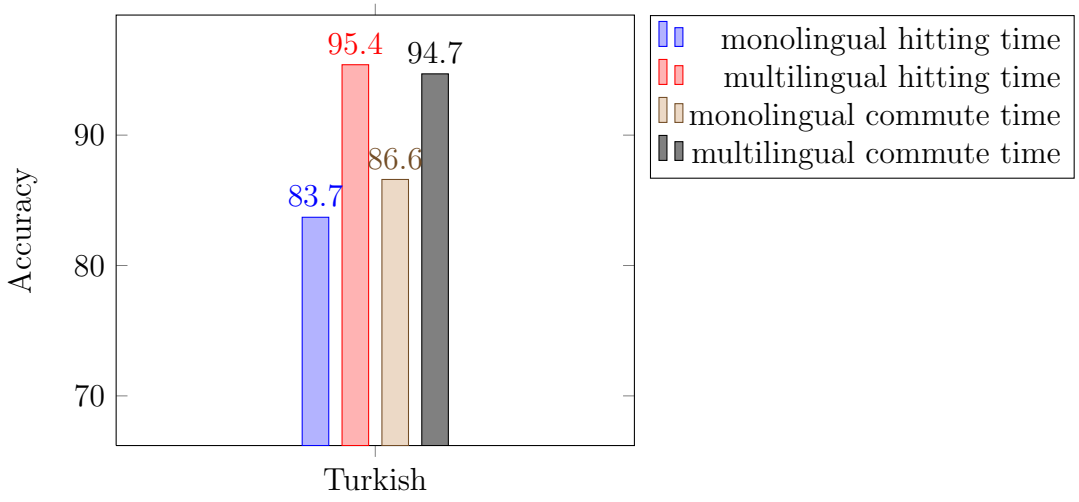


(b) T=30 and M=1000

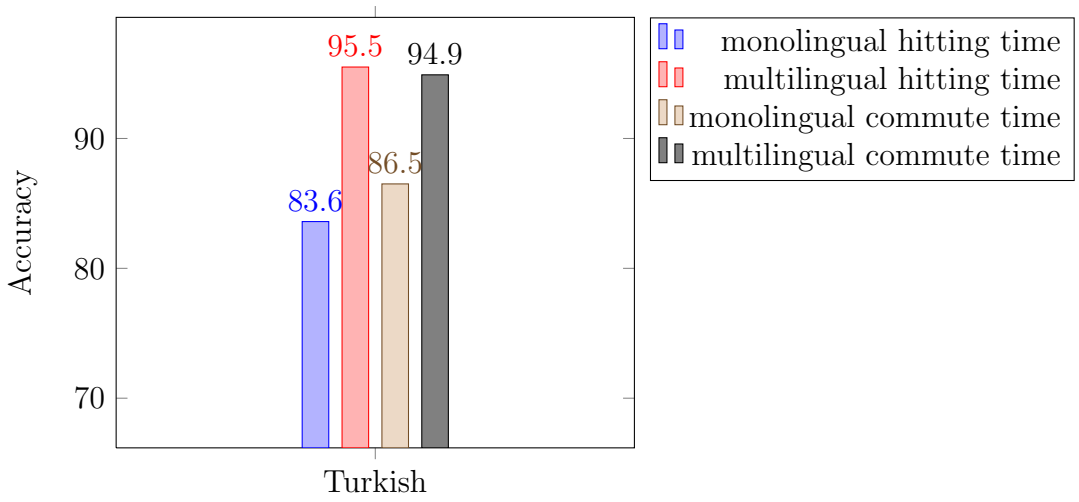


(c) T=45 and M=1000

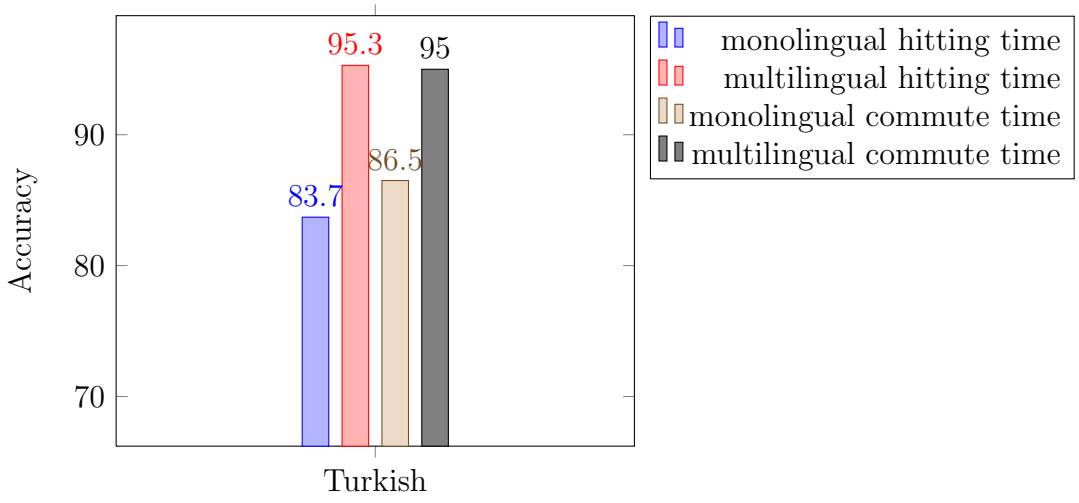
Figure 5.8. Multilingual accuracies using M=1000 and varying T for Turkish.



(a) T=30 and M=1000



(b) T=30 and M=2000



(c) T=30 and M=3000

Figure 5.9. Multilingual accuracies using T=30 and varying M for Turkish.

5.3. Varying Parameters

In previous sections, we compare hitting time and commute time for various combinations of M and T. Figure 5.10-13 illustrate the effect of varying these parameters on the performances of monolingual and multilingual algorithms.

It is shown in Figure 5.10 and Figure 5.11 that varying T and fixing M has a small effect on the performances of multilingual algorithms as well as monolingual algorithms on English and Turkish. In addition, varying M and fixing T also has a small effect on the performances of multilingual algorithms as well as monolingual algorithms as illustrated in Figure 5.12 and 5.13. All results under various M and T combinations are similar to each other. This suggests that multilingual and monolingual approaches are robust because they are not affected much by varying the parameters.

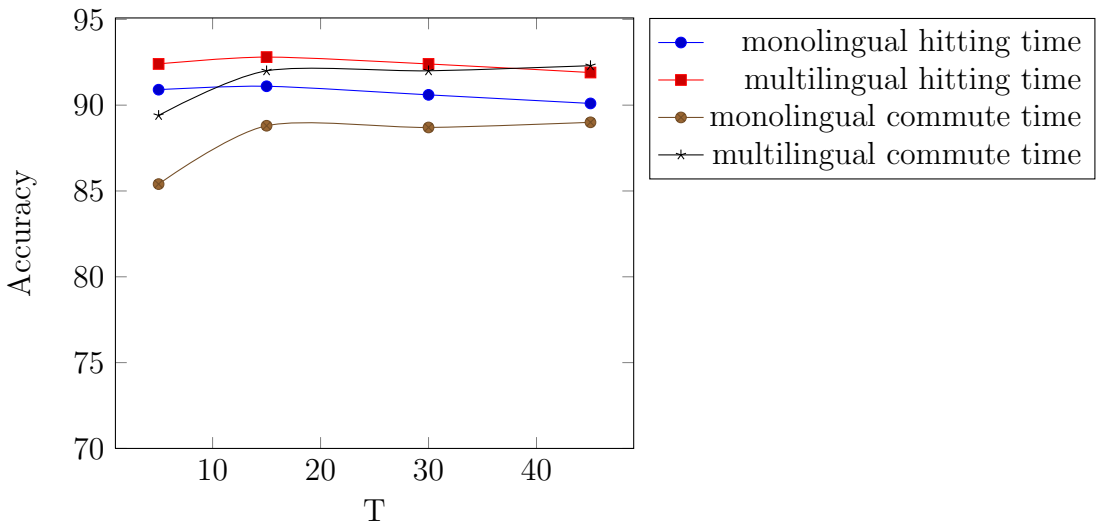


Figure 5.10. Effects of using M=1000 and varying T on English.

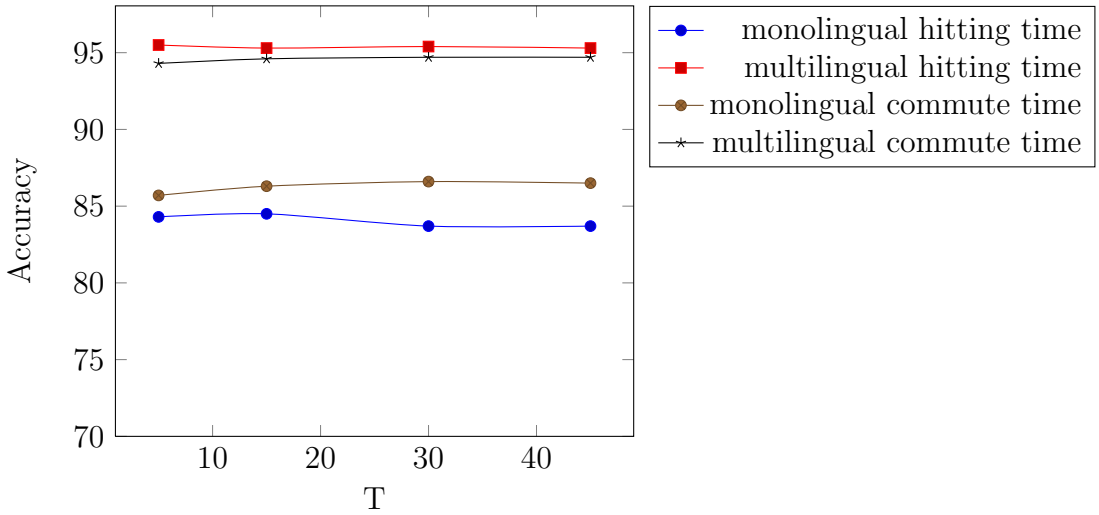


Figure 5.11. Effects of using M=1000 and varying T on Turkish.

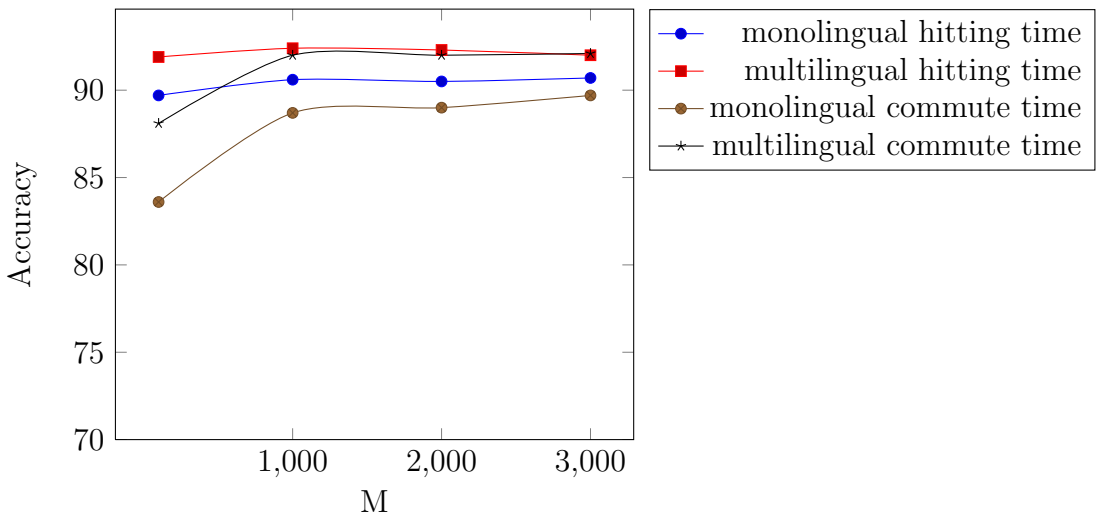


Figure 5.12. Effects of using T=30 and varying M on English.

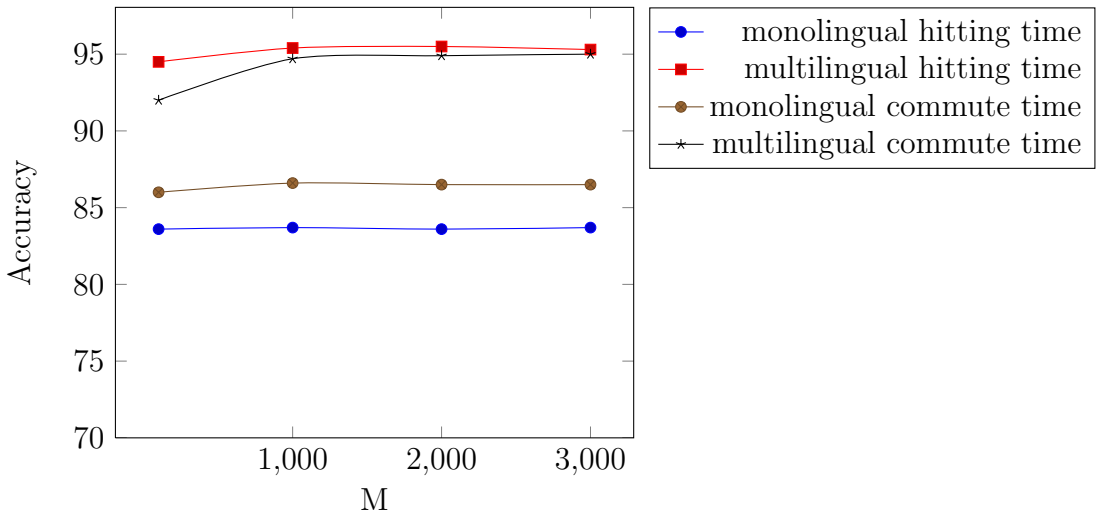


Figure 5.13. Effects of using T=30 and varying M on Turkish.

6. CONCLUSION

We addressed the problem of identifying the polarities of English and Turkish words. Most previous studies on polarity detection focus on English and depend on language specific resources such as WordNet. Many Non-English languages have WordNets. However, they are not as comprehensive as the English WordNet.

In this thesis, we develop an approach that utilizes the compatibility of English and Non-English WordNets to build a multilingual word relatedness graph. We propose using random walk model with commute time proximity measure over this graph to predict word polarities. We evaluate our approach for English and Turkish. We show that the random walk model with commute time achieves similar performance to the state of art method for English in the literature. Our multilingual approach based on connecting the English and Turkish word relatedness graphs led to significant improvement in performance for both languages. We achieved an accuracy of 92.8% for English and an accuracy of 95.5% for Turkish. To the best of our knowledge, we report the first word polarity detection results for Turkish. Our multilingual approach can be applied to other languages that have WordNets compatible with the English WordNet.

7. FUTURE WORK

In recent days, there is a trending issue Web 2.0 consisting of several online tools and platforms where people share their ideas, opinions and communicate with each other easily. Web 2.0 applications require more interaction with the end user. Web 2.0 increases use of blogs, micro-blogs, social networking websites and wikis. Therefore, the huge amount of data in Web 2.0 platforms can be an important resource for many organizations.

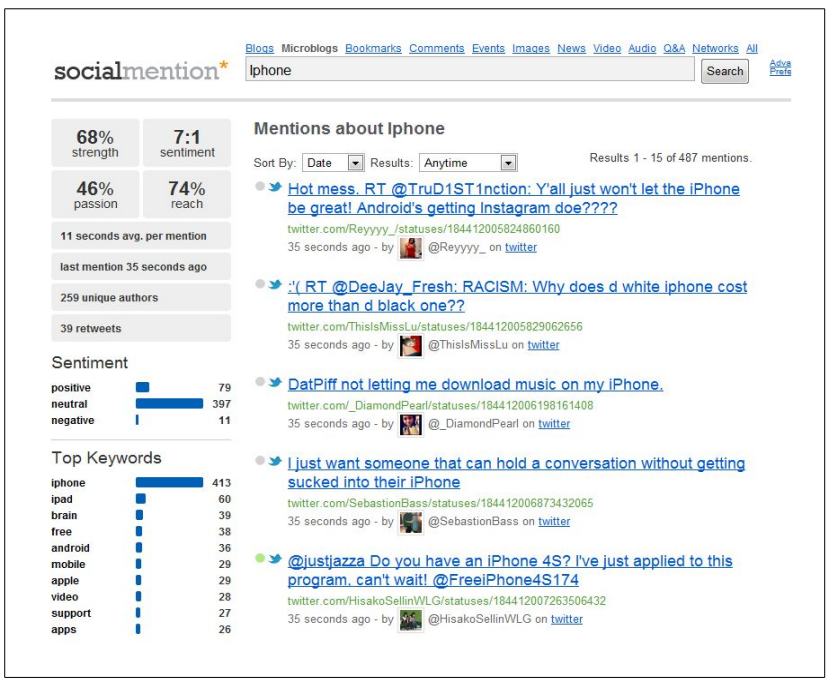


Figure 7.1. An example Web 2.0 application.

A growing number of companies are using Web 2.0 tools to improve communication with their consumers. They are trying to improve product development and service enhancement by using the feedbacks from end users. They control their process and strategies and change them if necessary. Therefore, analyzing the huge amount of data in Web 2.0 has an extreme importance in future's web.

One of the most popular areas in Web 2.0 is identifying text polarity. For instance, what is the consumer's attitude in a product review? What is the emotion of a person in a movie review? To be able to answer all of these questions, we must be able to

identify polarity of text. Socialmention.com, a screenshot of which is shown in Figure 7.1, is a popular realtime search engine for text polarity detection in English. You can search any keyword in Web 2.0 platforms (twitter, facebook, blogs etc.) by using SocialMention.com. It is shown in Figure 7.1 that the sentiment part of the search page shows how many of the search results are positive, negative and neutral. Another example of text polarity detection is tweetFeel.com which searches any keyword across various Web 2.0 platforms.

In contrast to English, Turkish does not have text polarity detection tools such as SocialMention.com. As we discussed before, since text consists of several words, word polarity detection is the first stage of text polarity detection. In this thesis, we developed word polarity detection system for Turkish. As future work, we aim to focus on text polarity detection. Another direction for research is extending our multilingual approach by making use of several other WordNets such as connecting Turkish, English and German WordNets. In addition, social media will gain more importance in future's web. Therefore, we believe that giving more priority to the linguistic studies related to text processing in Turkish is important. Text categorization, text summarization and information extraction are some of the important research directions for Turkish.

REFERENCES

1. Turney, P. D. and M. L. Littman, "Measuring Praise and Criticism: Inference of Semantic Orientation from Association", *ACM Transactions on Information Systems*, Vol. 21, No. 4, pp. 315-346, 2003.

2. Hatzivassiloglou, V. and K. R. McKeown, "Predicting the Semantic Orientation of Adjectives", *In Proceedings of the European Chapter of the ACL*, pp. 174-181, 1997.

3. Morinaga, D., K. Yamanishi, K. Tateishi and T. Fukushima, "Mining Product Reputations on the Web", *In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 341-349, 2002.

4. Popescu, A. and O. Etzioni, "Extracting Product Features and Opinions from Reviews", *In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing Association for Computational Linguistics*, pp. 339-346, 2005.

5. Turney, P. D., "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews", *In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 417-424, 2002.

6. Hassan, A., V. Qazvinian and D. Radev, "What's with the Attitude? Identifying Sentences with Attitude in Online Discussions", *In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 1245-1255, 2010.

7. Miller, G. A., "WordNet: A Lexical Database for English", *Communications of the ACM*, Vol. 38, No. 11, pp. 39-41, 1995.

8. Stone, P., D. Dunphy, M. Smith and D. Ogilvie, *The General Inquirer: A Computer Approach to Content Analysis*, The MIT Press, Cambridge, MA, USA, 1966.

9. Takamura, H., T. Inui and M. Okumura, "Extracting Semantic Orientations of Words Using Spin Model", *In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 133-140, 2005.

10. Hassan, A. and D. Radev, "Identifying Text Polarity Using Random Walks", *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 395-403, 2010.

11. Vossen, P., *Eurowordnet: A Multilingual Database with Lexical Semantic Networks*, Kluwer Academic Publishers, Norwell, MA, USA, 1998.

12. Tufiş, D., D. Cristea and S. Stamou, "Balkanet: Aims, Methods, Results and Perspectives. A General Overview", *Romanian Journal on Science and Technology of Information*, Vol. 7, pp. 9-43, 2004.

13. Stamou, S., K. Oflazer, K. Pala, D. Christoudoulakis, D. Cristea, D. Tufis, S. Koeva, G. Totkov, D. Dutoit and M. Grigoriadou, "Balkanet: A multilingual Semantic Network for Balkan Languages", *In Proceedings of the First International WordNet Conference*, pp. 12-14, 2002.

14. Bilgin, O., Ö. Çetinoglu and K. Oflazer, "Building a Wordnet for Turkish", *Romanian Journal on Information Science and Technology*, Vol. 7, pp. 163-172, 2004.

15. Bilgin, O., Ö. Çetinoglu and K. Oflazer, "Morphosemantic Relations In and Across Wordnets: A Preliminary Study Based on Turkish", *In Proceedings of the Second Global WordNet Conference*, 2004.

16. Lovasz, L., "Random Walks on Graphs: A Survey", *Bolyai Society Mathematical Studies*, Vol. 2, pp. 353-398, Budapest, 1996.

17. Kamps, J., M. Marx, R. J. Mokken and M. D. Rijke, "Using Wordnet to Measure Semantic Orientations of Adjectives", *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, pp. 1115-1118, 2004.

18. Hassan, A., A. Abu-Jbara, R. Jha and D. Radev, "Identifying the Semantic Orientation of Foreign Words", *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Vol. 2, pp. 592-597, 2011.

19. Vossen, P., "EuroWordNet: Linguistic Ontologies in a Multilingual Database", *Communication and Cognition for Artificial Intelligence*, Vol. 15, pp. 37-80, 1998.

20. Sarkar, P. and A. W. Moore, "A Tractable Approach to Finding Closest Truncated-commute-time Neighbors in Large Graphs", *The 23rd Conference on Uncertainty in Artificial Intelligence*, pp. 335-343, 2007.

21. Sarkar, P., *Tractable Algorithms for Proximity Search on Large Graphs*, Ph.D. Thesis, Carnegie Mellon University, 2010.