

ÇANKAYA UNIVERSITY  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
MATHEMATICS AND COMPUTER SCIENCE

MASTER THESIS

THE EFFECTS OF MORPHOLOGICAL STRUCTURE OF TURKISH ON  
SEMANTIC RELATEDNESS

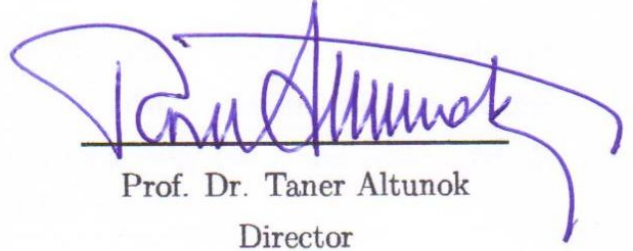
UĞUR SOPAOĞLU

July 2014


Title of the Thesis: **The Effects of Morphological Structure of Turkish on Semantic Relatedness**

Submitted by **Uğur SOPAOĞLU**

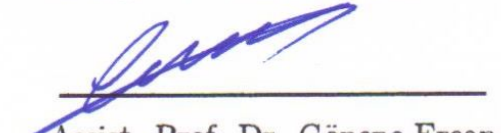
Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

  
Prof. Dr. Taner Altunok  
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science




  
Assist. Prof. Dr. Murat Saran  
Head of Department

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

  
Assist. Prof. Dr. Gönenç Ercan  
Supervisor

Examination Date: 24.07.2014

**Examining Committee Members**

Prof.Dr. Mehmet Reşit Tolun (Aksaray Univ.)   
Assist. Prof. Dr. Gönenç Ercan (Çankaya Univ.)   
Assist. Prof. Dr. Abdül Kadir Görür (Çankaya Univ.) 

## STATEMENT OF NON-PLAGIARISM PAGE

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

Name, Last Name : Uğur SOPAOĞLU

Signature :



Date :

24.07.2014

## ABSTRACT

### THE EFFECTS OF MORPHOLOGICAL STRUCTURE OF TURKISH ON SEMANTIC RELATEDNESS

SOPAOĞLU, Uğur

M.Sc., Department of Computer Engineering

**Supervisor:** Assist. Prof. Dr. Gönenç ERCAN

July 2014, 47 pages

It has been thought that the morphological analysis on agglutinative languages affects the success of semantic relatedness positively. In this study, semantic relatedness is tested to support this idea performing morphological analysis on Turkish. To understand the effect of morphology, the accomplishment of semantic relatedness is measured using two different methods, which are word association and clustering purity. According to results of these methods, it has been shown how much morphology affects semantic relatedness.

**Keywords:** Semantic Relatedness, Semantic Similarity, Effect of Morphology.

ÖZ

**TÜRKÇE'NİN MORFOLOJİK YAPISININ ANLAMSAL İLİŞKİ  
ÜZERİNDEKİ ETKİLERİ**

SOPAOĞLU, Uğur

M.Sc., Bilgisayar Mühendisliği Bölümü

**Tez Yöneticisi:** Assist. Prof. Dr. Gönenç ERCAN

Temmuz 2014, 47 pages

Eklemeli diller üzerinde morfolojik analizin, anlamsal ilişkinin başarısını olumlu etkileyeceği düşünülmüştür. Bu çalışmada, bu fikri doğrulayabilmek için Türkçe üzerinde morfolojik analiz yapılarak anlamsal ilişki test edilmiştir. Morfolojinin etkisini anlayabilmek için kelime ilişkilendirme ve kümeleme safılığı olmak üzere iki farklı yöntemle anlamsal ilişkinin başarısı ölçülmüştür. Bu yöntemlerin sonuçlarına göre morfolojinin anlamsal ilişkiyi ne kadar etkilediği deney sonuçlarında gösterilmiştir.

**Anahtar Kelimeler:** Anlamsal İlişki, Anlamsal Benzerlik, Morfolojinin Etkisi.

## ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my thesis advisor Assist.Prof. Dr. Gönenç ERCAN, who has encouraged and guided me throughout this thesis patiently.

I had a lot of useful discussions with Assist.Prof. Dr. Abdül Kadir GÖRÜR . I would like to thank him cordially for his valuable comments.

I wish to thank the examining committee for their kindness during the presentation of this thesis.

I would like to express my deep gratitude to my family for their endless and continuous encourage and support throughout the years.

Finally, I would like to thank Engin ÖZBEY with whom we have shared good and bad times for many years.

## TABLE OF CONTENTS

STATEMENT OF NON-PLAGIARISM .....	iii
ABSTRACT .....	iv
ÖZ.....	v
ACKNOWLEDGMENTS .....	vi
TABLE OF CONTENTS .....	viii
LIST OF TABLES .....	ix
LIST OF FIGURES.....	x

### CHAPTERS:

<b>1. INTRODUCTION.....</b>	<b>1</b>
<b>2. BACKGROUND AND RELATED WORKS .....</b>	<b>4</b>
<b>2.1. Semantic Relatedness.....</b>	<b>4</b>
<b>2.2. Related Works .....</b>	<b>4</b>
<b>2.2.1. Semantic Relatedness .....</b>	<b>4</b>
<b>2.2.2. Morphological Parser and Disambiguation .....</b>	<b>5</b>
<b>2.3. Semantic Relatedness Methods.....</b>	<b>6</b>
<b>2.3.1. Knowledge Based Semantic Relatedness Measures .....</b>	<b>6</b>
<b>2.3.1.1. Knowledge Based Semantic Relatedness Methods...</b>	<b>10</b>
<b>2.3.2. Corpus Based Semantic Relatedness Measures.....</b>	<b>13</b>
<b>3. METHODS .....</b>	<b>17</b>
<b>3.1. Measurements of Semantic Relatedness .....</b>	<b>17</b>
<b>3.1.1. Parsing Corpus .....</b>	<b>17</b>
<b>3.1.2. Building Dictionary .....</b>	<b>18</b>
<b>3.1.3. Building Co-occurrence Matrix .....</b>	<b>19</b>
<b>3.1.4. Weighing the Co-occurrence Matrix.....</b>	<b>20</b>
<b>3.1.5. Dimension Reduction Using SVD .....</b>	<b>21</b>
<b>3.2. Morphological Parser and Disambiguator.....</b>	<b>22</b>
<b>3.2.1. Morphological Parser .....</b>	<b>22</b>
<b>3.2.1.1. Snowball.....</b>	<b>22</b>
<b>3.2.1.2. Zemberek .....</b>	<b>23</b>
<b>3.2.1.3. Morphological Parser.....</b>	<b>24</b>

<b>3.2.2.</b> Morphological Disambiguator .....	25
<b>3.2.2.1.</b> Yüret and Türe Morphological Disambiguator .....	25
<b>3.2.2.2.</b> Hasim Sak Method .....	27
<b>4.</b> EXPERIMENTS and RESULTS .....	29
<b>4.1.</b> Experiments .....	29
<b>4.1.1.</b> Clustering Purity .....	29
<b>4.1.2.</b> Word Association .....	30
<b>4.2.</b> Results .....	32
<b>4.2.1.</b> Effects of Morphology on Semantic Relatedness .....	32
<b>4.2.1.1.</b> No-Stemmer Experiment .....	32
<b>4.2.1.2.</b> Snowball Experiment .....	33
<b>4.2.1.3.</b> Zemberek Experiment .....	35
<b>4.2.1.4.</b> Hasim Sak Morphological Parser .....	37
<b>4.2.1.5.</b> Hasim Sak Morphological Disambiguator .....	38
<b>4.2.2.</b> The Effect of Dimension Reduction .....	43
<b>5.</b> CONCLUSION AND FUTURE WORK .....	46
<b>5.1.</b> Future Work .....	47
REFERENCES .....	R1



## LIST OF TABLES

<b>Table 1</b>	The Output Sample of Semantic Relatedness.....	2
<b>Table 2</b>	English WordNet 3.0 [56] .....	8
<b>Table 3</b>	WordNet for Other Languages [4] .....	8
<b>Table 4</b>	Statistical Information about Wikipedia at 27 January 2014 [59].....	9
<b>Table 5</b>	Results of The TOEFL and ESL Questions.....	16
<b>Table 6</b>	The Number of Unique Terms.....	19
<b>Table 7</b>	The Number of Unique Terms After Frequency Filter.....	20
<b>Table 8</b>	Other Language Snowball Stemmer .....	23
<b>Table 9</b>	Turkish Snowball Examples [14] .....	23
<b>Table 10</b>	Morphological Parser Results .....	26
<b>Table 11</b>	Parses of “Masali” [63] .....	27
<b>Table 12</b>	Disambiguation Results .....	28
<b>Table 13</b>	Two Sample Categories .....	30
<b>Table 14</b>	Information about to be applied tests .....	31
<b>Table 15</b>	No-Stemmer Word Association Experiment .....	32
<b>Table 16</b>	Snowball Stemmer Word Association Experiment .....	34
<b>Table 17</b>	Zemberek Parser Experiment Semantic Relatedness Results .....	35
<b>Table 18</b>	Hasim Sak Parser Word Association Experiment .....	37
<b>Table 19</b>	Hasim Sak Disambiguator Word Association Experiment .....	39
<b>Table 20</b>	Disambiguator Experiment For First Derivational Affix .....	40
<b>Table 21</b>	Disambiguator Experiment For Last Derivational Affix .....	42
<b>Table 22</b>	SVD Truncate Value is 200 for MD .....	43
<b>Table 23</b>	SVD Truncate Value is 600 for MD .....	44

## LIST OF FIGURES

<b>Figure 1</b>	The sample of WordNet .....	7
<b>Figure 2</b>	Wu&Palmer method [61].....	11
<b>Figure 3</b>	Sample of ESL exam .....	16
<b>Figure 4</b>	Sample of Toefl exam.....	16
<b>Figure 5</b>	The sample of WordNet .....	19
<b>Figure 6</b>	No-Stemmer word association experiments.....	33
<b>Figure 7</b>	No-Stemmer clustering purity experiments.....	33
<b>Figure 8</b>	Snowball word association experiments .....	34
<b>Figure 9</b>	Clustering purity for snowball experiments.....	35
<b>Figure 10</b>	Zemberek word association experiments.....	36
<b>Figure 11</b>	Zemberek clustering purity experiments.....	36
<b>Figure 12</b>	Hasim Sak parser word association experiments.....	38
<b>Figure 13</b>	Hasim Sak parser clustering purity experiments.....	38
<b>Figure 14</b>	Hasim Sak disambiguator word association experiments .....	39
<b>Figure 15</b>	Hasim Sak disambiguator clustering purity experiments .....	40
<b>Figure 16</b>	Hasim Sak disambiguator experiments for first derivational affix....	41
<b>Figure 17</b>	Clustering purity experiments for first derivational affix .....	41
<b>Figure 18</b>	Hasim Sak disambiguator experiments for last derivational affix....	42
<b>Figure 19</b>	Clustering purity experiments for last derivational affix .....	42
<b>Figure 20</b>	SVD truncate value $k$ is 200.....	44
<b>Figure 21</b>	SVD truncate value $k$ is 600.....	45

## LIST OF ABBREVIATIONS

NLP	Natural Language Processing
SR	Semantic Relatedness
SVD	Singular Value Decomposition
SS	Semantic Similarity
MP	Morphological Parser
MD	Morphological Disambiguator
SSA	Salient Semantic Analysis
ESA	Explicit Semantic Analysis
KBSR	Knowledge Based Semantic Relatedness
SOC-PMI	Second Order Pointwise Mutual Information
PMI	Pointwise Mutual Information
LSA	Latent Semantic Analysis
PMI-IR	Pointwise Mutual Information and Information Retrieval
TOEFL	Test of English as a Foreign Language
ESL	English as a Second Language
PMI	Pointwise Mutual Information
XML	Extensible Markup Language
FSM	Finite State Machine
GPA	Greedy Prepend Algorithm

## CHAPTER 1

### INTRODUCTION

Semantic Relatedness can be defined as semantic closeness of two words or two concepts. The closeness involves all relations such as synonym, antonym, *is – a* or *has – a* relation. For example, there is a globally known relation between car and motorcycle whereas there is no relation between car and library.

This is a popular research area because it helps to solve different Natural Language Processing (NLP) problems. A lot of study [51, 40, 19, 7] is done until now. However, only one study [11] is performed on Turkish about Semantic Relatedness (SR).

Bullinaria study tests the effects of stems on English but Turkish is an agglutinative language so morphological structure of a word can affect the accomplishment of SR so this thesis focuses on the effect of morphology on SR for Turkish. Ercan study [11] examines effect of morphology simply but we can not decide the effect of morphology on SR using the result of Ercan study for Turkish. In this study, Turkish morphology is examined with all details for the effect of morphology on semantic relatedness. In addition, the result of this study provides an idea about the effect of morphology for other agglutinative languages.

Wikipedia is identified as a corpora to perform this study. Co-occurrence statistics of words in the corpus are produced using the Wikipedia articles. Latent Semantic Analysis method is used to calculate semantic relatedness. When SR is calculated, two different types of experiments are performed on the corpora. Namely Word Association and Clustering Purity

According to these types,

1. First experiment tests the effect of morphology on SR. While the experiment is being performed, six different morphological processing techniques are tested which are as follows:

- In the first experiment, SR is calculated according to the words in Wikipedia without any modification.
  - In the second experiment, SR is calculated according to the words in Wikipedia but inflections of the words are removed.
  - In the third experiment, Zemberek is used to detect the roots of words in Wikipedia and SR is calculated according to these roots.
  - In the fourth experiment, Hasim Sak morphological parser is used to detect the roots of words in Wikipedia and SR is calculated according to these roots.
  - In the fifth experiment, Hasim Sak morphological disambiguator tool is used to decide which meaning of the word is used in the sentence and the root of the word is identified by morphological disambiguator according to the meaning of the word. SR is calculated according to these roots.
  - In the last experiment, the effect of Singular Value Decomposition (SVD) is tested. The experiment is performed using three different truncated value parameters.
2. In the second experiments, words are categorized according to the results of semantic relatedness values and the accomplishment of this experiments is evaluated.

Word 1	Word 2	Human Judgement	System Score
serf	köle	3.6136	0.2719
sihirbaz	büyücü	4.2272	0.19615
parça	bütün	4.090	0.1280
silah	çorap	1.2727	0.0231
siyaset	futbol	1.8863	0.1055
yolculuk	seyahat	4.9545	0.5458
vinç	alet	3.3863	0.0551
sığın	kabristan	1.25	-0.0306
fırın	ocak	4.7272	0.2490
Correlation	0.7699		

Table 1: The Output Sample of Semantic Relatedness

In Table 1, small part of the first experiment is shown. In the example, word pairs are compared with each other according to the semantic relatedness, also people judged the word pair and gave score between 1 and 5. In Table 1 first value shows average of human judgement scores and second value is assigned by program. Finally, the correlation between given people scores and program scores is calculated. The sample calculated correlation is seen at the end of the output.

In the clustering purity experiment, CLUTO [27] tool is used to categorize the words. When the categorization is performed for word set, k way clustering algorithm is used.

This thesis is structured as follows:

In **Chapter 2**, necessary background information and related works about the semantic relatedness are given. Important studies about semantic relatedness are described. In addition, difference between morphological parser and disambiguator are explained and related works to morphological disambiguator and parser are outlined. Additionally, corpus based and knowledge based semantic relatedness methods are explained in this chapter.

In **Chapter 3**, methods used to measure the semantic relatedness are explained step-by-step in detail.

In **Chapter 4**, all experiments and parameters of these experiments are defined. The results of these experiments are shown in Chapter 4. Different algorithms and parameters are compared with each other according to their results.

In **Chapter 5**, the results are evaluated and the most suitable parameters and form of morphological structure for accomplishment of SR are identified. Furthermore, what can be done in the future about the semantic relatedness is shown in Chapter 5.

## CHAPTER 2

### BACKGROUND AND RELATED WORKS

#### 2.1 Semantic Relatedness

Semantic Relatedness (SR) measures the degree and strength of semantic relations between concepts, while these relations can be classical such as synonymy, antonymy, hyponymy, meronymy, holonymy and hypernymy [51, 40, 19] they can also be non-classical. On the other hand, Semantic Similarity (SS) decides how similar the meanings of two words are to each other [53]. SS actually is a special case of SR. SS and SR might be confused, for example; whereas **house** and **cabin** are similar word pairs; **saddle** and **bicycle** are related word pairs. SR and SS measurements are widely used in many Natural Language Processing (NLP) systems and tasks such as word sense disambiguation [39, 38], text summarization [6], keyword extraction [66], assessing topic coherence [34], information retrieval [58], automatic correction of word errors, which are called malapropism, in documents [21]. Researchers show an interest in SR and SS because they play an important role in increasing accomplishment on NLP systems [62].

#### 2.2 Related Works

##### 2.2.1 Semantic Relatedness

There has been an increase in the number of studies done in this area [51, 40, 22, 65]. However many of them are focusing on English. Houghes et al. [22] have measured lexical semantic relatedness using Random Walk and Markov Chain Theory. Correlation of WordNet and human judgement are calculated as 0.9. Salient Semantic Analysis (SSA) has been developed to measure SR by analysing the link on the documents [19]. Zesch et al. [65] computes SR using Wiktionary and have compared the performance of Wiktionary with WordNet and GermaNet

[18]. Satanjeev and Pedersen [5] measures SR benefit from the overlaps of definition of the words . Unlike a lot of other studies, in the Zesch and Gurevych research [64], they use automatically created dataset to calculate the SR whereas other studies generally use manually created dataset. The dataset consists of word pairs and the importance of the these word pairs contains all lexical relations and this is important to measure SR.

In recent semantic analysis methods, Wikipedia is preferred as a corpus [51, 22, 15]. In the Ponzetto and Strube study [41], they calculate SR using Wikipedia and WordNet on different datasets. They obtain that the result of Wikipedia is more successful than the result of WordNet. Eric et al. [62] have created a graph using the hyperlinks between articles in Wikipedia. The graph provides a large amount of information about the relationships between articles. Eric et al. research used random walk method on the graph and existing Wikipedia based studies improved using this technique. Gabrilovich et al. [15] propose a novel method called as Explicit Semantic Analysis (ESA). In the ESA, correlation of human judgement and scores produced by this method are calculated as 0.75 for word pairs and also in this study, Wikipedia has been used as corpus.

To the best of my knowledge for the Turkish language, only one study exists [11] which used Wikipedia as a raw text corpora to measure SR. This thesis has been based on the study conducted on Turkish SR. However Ercan [11] research does not investigate morphological form in detail.

### **2.2.2 Morphological Parser and Disambiguation**

Morphological Parser (MP) separates a word into its morphemes, for example in English, “changelessness” is decomposed to “change-less-ness”. MP is an important step for many NLP systems. It may be used in text-to-speech conversion systems [20], speech recognition systems [3], machine translation and question answering systems [42]. There are a lot of studies about MP [25, 45, 2] and for Turkish [49, 12, 35]. Morphological Disambiguator (MD) selects the proper parse of the word according to the sentence. The studies about MD can be divided into two groups: statistical approaches [17, 16], rule based approaches [36].

Bullinaria and Levy study [7] research investigate different parameters regarding stop lists, stemming and dimensionality reduction on the performance of semantic



relatedness for English language. We performed a similar experiment for Turkish language since Turkish is an agglutinative language the effect of the morphological processing variations can be expected to have a greater impact.

## **2.3 Semantic Relatedness Methods**

Semantic Relatedness identifies the level of the relationship between two terms. Generally SR methods return a number which represent the level of relationship. Magnitude of the return value shows the strength of the relation. Lower values show that these terms are not related whereas higher values show that these terms are related with each other. While the value of SR is being calculated, many factors may cause the findings to deviate from the human judgement such as lack of the size of corpus, sense disambiguation problem etc.

SR has succeeded in attracting the attention of researchers by virtue of the wide usage area. However there is only one study about SR on Turkish [11]. In this study, words have been added to the dictionary with the roots of words. How the value of SR is affected by the insertion of the words to the dictionary according to their morphological structure has been studied in this thesis. The results are expected to be successful as Turkish is an agglutinative language. Turkish words consist of a root and many other inflections and derivational affixes. In this study, we have tested three different morphological forms. The first morphological state is the root of the word. The second morphological state is root and all inflections up to the first derivational affix inclusively. The last one is root and all inflections up to the last derivational affix inclusively. In this thesis, these three morphological states have been inserted to the dictionary which contains all unique word on the corpus.

SR methods can be divided into two groups: knowledge based and corpus based [19].

### **2.3.1 Knowledge Based Semantic Relatedness Measures**

Information is extracted from resources which are constructed manually by humans in Knowledge Based. The most known Knowledge Based Semantic Relatedness (KBSR) resources are WordNet, Wikipedia and Wiktionary.

## WORDNET

WordNet [31] is a lexical Thesaurus constructed manually by linguists. WordNet consists of words and their senses. Each word can have multiple senses and there may also be more than one word with the same meaning.

Figure 1 shows the WordNet senses for the word "intelligence" and also shows the synonyms in parentheses.

- **(intelligence)** the ability to comprehend; to understand and profit from experience.
- **(intelligence, intelligence service, intelligence agency)** a unit responsible for gathering and interpreting information about an enemy.
- **(intelligence, intelligence information)** secret information about an enemy.
- **(intelligence, news, tidings, word)** information about recent and important events.
- **(intelligence, intelligence activity, intelligence operation)** the operation of gathering information about an enemy.

Figure 1: The sample of WordNet

Additionally, one of the most important features of WordNet is that it keeps relations between word senses. WordNet provides convenience to classical semantic relationships between two words. Some relations are as follows:

- **Synonym:** "a word or phrase that has the same or nearly the same meaning as another word or phrase in the same language" <sup>1</sup>
- **Antonym:** "a word that means the opposite of another word" <sup>1</sup>
- **Hyponymy:** "a word that is more specific than a given word" <sup>2</sup>
- **Meronymy** "the semantic relation that holds between a part and the whole"<sup>2</sup>

---

<sup>1</sup> Description has been taken from Cambridge Dictionary

<sup>2</sup> Description has been taken from WordNet

- **Troponymy** “the semantic relation of being a manner of does something whole”<sup>3</sup>
- **Holonymy** “the semantic relation that holds between a whole and its parts”<sup>3</sup> (Holonymy is the opposite of meronymy.)
- **Hypernymy** “the semantic relation of being superordinate or belonging to a higher rank or class”<sup>3</sup>

Table 2 shows some statistical information about WordNet for English and Turkish:

Pos	Unique String	Synsets	Total Word-Sense Pairs
Noun	117798	82115	146312
Verb	11529	13767	25047
Adjective	21479	18156	30002
Adverb	4481	3621	5580
Totals	155287	117659	206941

Table 2: English WordNet 3.0 [56]

Pos	Bulgarian	Czech	Greek	Romanian	Turkish	Serbian
Noun	14174	21009	14426	13345	11059	8059
Verb	4169	5155	3402	4808	2725	1803
Adjective	3088	2128	617	852	802	324
Adverb	9	164	16	834	40	13
Synsets	21441	28456	18461	19839	14626	8059

Table 3: WordNet for Other Languages [4]

When the two tables above are examined, WordNet for English appears to be more comprehensive than WordNet for other languages. In addition, in the English WordNet, the number of words are ten times bigger than WordNet for Turkish.

WordNet identifies semantic relations. To calculate semantic relationship different methods[5, 61] are used but there are components in Turkish WordNet sense graph. These components make some of the semantic relatedness measures invalid for Turkish.

---

<sup>3</sup> Description has been taken from WordNet

Studies have been developed using WordNet on the NLP systems and related areas [10, 29, 21, 26, 43]. However, it can be stated that there are some deficiencies of WordNet. For example, WordNet 2.1 does not include sufficient named entities [51].

Furthermore, Roget's Thesaurus is an another resource similar to WordNet. Roget's Thesaurus is richer than WordNet on the basis of relationships (IS-A or HAS-A) between words [30].

## WIKIPEDIA

Wikipedia is an online encyclopedia that contains vast amount of semi-structured data in its articles about various topics in 287 different languages. Wikipedia provides service to users since 15 January 2001. According to the comScore firm, Wikipedia, which comprises 18 billion web pages, is visited by approximately 500.000 unique visitors per month [9]. Table 4 shows some statistical information about Wikipedia in some languages.

Language	Articles	Total	Admins	Users	Images
English	4,461,108	32,313,108	1,418	20,821,464	826,066
Dutch	1,763,378	3,176,223	53	583,132	19
German	1,692,138	4,694,005	255	1,822,849	162,441
Swedish	1,612,177	3,590,807	73	367,964	0
French	1,481,067	6,338,668	182	1,765,781	42,625
Italian	1,102,671	3,592,277	108	981,912	125,269
Turkish	224,608	1,114,268	28	548,956	28,005

Table 4: Statistical Information about Wikipedia at 27 January 2014 [59]

The table shows 6 columns which are Language, Articles, Total, Admins, Users and Images referring to the languages of article and number of articles, pages in all namespaces, admin users, registered users, and uploaded images respectively. According to the statistics, English Wikipedia draws the attention of researchers due to its completeness and size as a comprehensive encyclopedia. Furthermore, Wikipedia offers a decent corpus for researcher who want to study under resourced languages such as Turkish.

Semantic relatedness researchers are aware of provided opportunities by Wikipedia. Hence, they have been using Wikipedia on their researches [51, 54, 15, 41, 62, 60].

While SR is being calculated, many features of Wikipedia is used in the study such as redirect pages, disambiguation pages and internal links.

Strube and Ponzetto compare SR methods using WordNet and Wikipedia [51] and shows that Wikipedia achieves better results than WordNet. In addition, WordNet is not as vast as Wikipedia in terms of named entity.

### 2.3.1.1 Knowledge Based Semantic Relatedness Methods

#### L&C Method

Leacock and Chodorow [57] propose a Word Sense Identification method. While determining the sense of the word, to create an effective training set, they use WordNet features. For example, WordNet contains a lot of polysemous words and semantic relations. WordNet’s IS-A relation feature is used to measure the similarity between two words. The semantic similarity is calculated, using the following equation:

$$sim(w_1, w_2) = max \left[ -log\left(\frac{N_p}{2D}\right) \right] \quad (2.1)$$

Where  $N_p$  is the number of nodes between two words ( $w_1$  and  $w_2$ ),  $D$  is the maximum length of the shortest path between any two words in taxonomy.

#### Wu&Palmer Method

Wu & Palmer method [61] translates the verbs in two languages, namely, English and Chinese. Wu & Palmer propose a novel method in which a lot of different concepts are identified for each verb to choose the correct verb to be translated. The similarity is measured between concepts using the equation:

$$ConceptSimilarity(C_1, C_2) = \frac{2 * N_3}{N_1 + N_2 + 2 * N_3} \quad (2.2)$$

Where,  $C_1$  and  $C_2$  are concepts and  $C_3$  is a concept and it is the closest common ancestor of  $C_1$  and  $C_2$ .  $N_1$  is the number of nodes between  $C_1$  and  $C_3$ ,  $N_2$  is the length of the shortest path between  $C_2$  and  $C_3$  and  $N_3$  is the number of nodes between  $C_3$  and ROOT.

Semantic Relatedness is directly proportional to the depth of  $c_3$  between common ancestor of two concepts and ROOT and closeness of  $c_1$  and  $c_2$  to  $c_3$ .

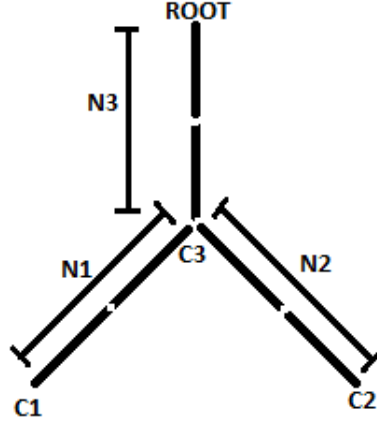


Figure 2: Wu&Palmer method [61]

After the method is applied to the dataset to measure similarity, verb selection shows an increase of 13.8%. Whereas the result at the beginning of the study had been observed as 75%, the result at the end of the study shows progress. Final result is observed as 88.8%

### Resnik Method

In the Resnik's research [43] conceptual similarity is calculated using the path length between concepts on WordNet graph. In WordNet, each sense is considered as a node.

Resnik calculate the probability of each node as equation below according to the occurrence of concepts.

$$sum(g_i) = \sum_{g_k \in children(g_i)} sum(g_k) \quad (2.3)$$

$$prob(g_i) = \frac{sum(g_i)}{N} \quad (2.4)$$

In the equation 2.3,  $g_i$  is a node,  $g_k$  is a child of  $g_i$ . The number of occurrence of any node is calculated as the summation of the number of occurrence of the children of the node and the number of occurrence of the node.

In the equation 2.4, N is the number of words in the corpus.

Experimental results show that the method is more successful than traditional edge counting method in measurement of semantic similarity[43]. However, the shortcoming of this research is that it neglects the possible variety (hypernym

and hyponym) of the distance between links. While the conceptual similarity is calculated, following formula is used:

$$sim(g_i, g_j) = -\log(p(lso(g_i, g_j))) \quad (2.5)$$

where, lso is lowest common ancestor of the concepts

## J&C Method

This study is developed on the deficiencies in the research of Resnik by Jiang and Conrath [24]. Resnik's study uses the node based approach and downplays the effect of link distance.

In the Jiang and Conrath[24] study, the effect of edges can not be ignored on the semantic similarity measurement. In the Jiang study, semantic similarity is measured using the network edges and corpus statistics. In the IS-A hierarchy, semantic distance is computed using the following equation:

$$p(c|par(c)) = \frac{p(c \& par(c))}{p(par(c))} \quad (2.6)$$

Where,  $c$  is the child concept,  $par(c)$  is the parent concept of the child and the function  $p$  refers to probability. The probability of occurrence of any concept  $c$  and its parent  $par(c)$  is equal to  $p(c)$  thus  $p(c \& par(c)) = p(c)$ . If this is applied to Equation 2.6, it takes the form:

$$p(c|par(c)) = \frac{p(c)}{p(par(c))} \quad (2.7)$$

taking the logarithm of Equation 2.5 results in:

$$dist(c, par(c)) = -\log(p(c|par(c))) = \log(p(par(c))) - \log(p(c)) \quad (2.8)$$

Effect of other factors such as node depth, edge density can be integrated as in the equation below:

$$weight(c, par(c)) = \left( (\beta + (1 - \beta) \frac{\bar{E}}{E(par(c))}) \left( \frac{d(par(c)) + 1}{d(par(c))} \right)^\alpha \right) weight(c, par(c)) \quad (2.9)$$

Where,  $\bar{E}$  shows the average number of edges.  $\alpha$  and  $\beta$  control the weight of node depth and edge density.

## 2.3.2 Corpus Based Semantic Relatedness Measures

### Second Order Pointwise Mutual Information

Second Order Pointwise Mutual Information (SOC-PMI) is a corpus based approach used to measure the similarity between two words [23]. In SOC-PMI, the PMI value of neighbours of the target word is calculated and for each word the important neighbours are determined using the calculated PMI values.

While SOC-PMI is being calculated, following steps are applied:

1. First of all the number of words on the left and right side of a word decides. A sliding window of size ( $\nu$ ) is passed through the text, where the occurrence count of words in the same window is tracked.
2. PMI is calculated using the following equation

$$f_{pmi}(t_i, W) = \log_2 \frac{f_b(t_i, W) \times m}{f_t(t_i) f_t(W)} \quad (2.10)$$

where,  $f_t(t_i)$  (typed frequency) function is that the number of  $t_i$  appear in the corpus,  $f_b(t_i, W)$  (bigram frequency) function is that the number of  $t_i$  in the specified windows size,  $W$  is the unique word list,  $m$  is the total number of words and  $t$  means each unique word in the corpus.  $f_{pmi}$  is pointwise mutual information function. The PMI value of neighbours of each word are calculated and all neighbours are sorted in descending order according to the PMI value.

3. The first  $g$  words are taken from this sorted word lists.  $g$  is identified by using the following equation:

$$g_i = (\log(f_t(W_i)))^2 \frac{\log_2(n)}{\delta} \quad (2.11)$$

In the equation above, the value  $\delta$  is decided according to the size of corpus. In SOC-PMI study [23],  $\delta$  is accepted as 6.5 .

4. While two words are being compared,  $g$ -PMI summation should be calculated using the following equation:

$$f_g(W_1) = \sum_{i=1}^{g_1} (f_{pmi}(X_i, W_2))^\gamma \quad (2.12)$$



$$f_g(W_2) = \sum_{i=1}^{g_2} (f_{pmi}(Y_i, W_1))^\gamma \quad (2.13)$$

In the equations above, first equation calculates the  $g$ -PMI summation value for the first word ( $W_1$ ).  $X_i$  refers to the neighbour of the  $W_1$ . The second formula calculates  $g$ -PMI summation value for the second word ( $W_2$ ).  $Y_i$  means the neighbour of the first word.  $\gamma$  is the exponential parameter and  $n$  is the number of types.

5. Last step is the calculation of semantic similarity between two words:

$$Sim(W_1, W_2) = \frac{f_\beta(W_1)}{\beta_1} + \frac{f_\beta(W_2)}{\beta_2} \quad (2.14)$$

## Latent Semantic Analysis

Latent Semantic Analysis (LSA) is a corpus based model used to measure similarity between two words [28]. Steps of LSA is explained below:

1. The first step of LSA identifies the corpus to examine the similarity of word pairs.
2. The corpus is divided into text parts such as sentences, paragraphs, articles and etc.
3. Matrix is created using the corpus. In the matrix, each row represents the unique word in the corpus and each column represents a text. Each cell shows the frequency of the word corresponding part.
4. The next step aims at reducing the size of matrix by decomposing the created matrix in the previous step using Singular Value Decomposition (SVD) method. Detailed information on SVD is also provided in chapter 3.1.5.
5. Finally, similarity is calculated using the cosine similarity between two vectors. Cosine similarity equation is given below:

$$similarity(\vec{A}, \vec{B}) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (2.15)$$

In the equation above,  $\vec{A}$  and  $\vec{B}$  represent the vector which procure from SVD.

### PMI-IR Method

Pointwise Mutual Information and Information Retrieval (PMI-IR) is an unsupervised learning algorithm to identify the degree and strength of the similarity between words [55]. The method was tested on the Test of English as a Foreign Language (TOEFL), which consists of 80 synonym questions, and English as a Second Language (ESL), which consists of 50 synonym questions. The results of the method accomplish 73.75% for TOEFL and 74% for ESL. This method is developed based on conditional probability.

$$score(choice_i) = p(problem|choice_i) = \frac{p(choice_i \cap problem)}{p(choice_i)} \quad (2.16)$$

In the method, four different scenarios are evaluated using the AltaVista Advanced Search queries. These scenarios are as follow:

**Scenario 1:** In the first scenario, two words appear on the same document. In the equation below, hit refers to the number of document retrieved.

$$score_1(choice_i) = \frac{hits(problem \text{ AND } choice_i)}{hits(choice_i)} \quad (2.17)$$

**Scenario 2:** Two words appear on the same document within range of 10 words. In the equation below, NEAR is identified as a constraint and it means that two words are close to each other.

$$score_2(choice_i) = \frac{hits(problem \text{ NEAR } choice_i)}{hits(choice_i)} \quad (2.18)$$

**Scenario 3:** In the previous two scenarios, if two words are antonyms, they might score high as synonym. For this reason, following equation decreases the score of antonyms.

$$score_3(choice_i) = \frac{hits((prob \text{ NEAR } choice_i) \text{ AND } NOT((prob \text{ OR } choice_i) \text{ NEAR "not"}))}{hits(choice_i \text{ AND } NOT (choice_i \text{ NEAR "not"}))} \quad (2.19)$$

**Scenario 4:** The scenario actually is used for ESL exam because there is no context in the TOEFL whereas in the ESL exam, the context is provided to find which word can be used in the blank. Following equation solves this problem:

$$score_4(choice_i) =$$

$$\frac{\text{hits}((\text{prob NEAR choice}_i)\text{AND contextAND NOT}((\text{prob OR choice}_i)\text{ NEAR "not"}))}{\text{hits}(\text{choice}_i\text{ ANDcontext AND NOT} (\text{choice}_i\text{ NEAR "not"}))}$$

(2.20)

Table below shows the details of all scenarios for TOEFL and ESL.

Exam	Number Of Question	Scenario 1	Scenario 2	Scenario 3	Scenario 4
TOEFL	80	62.5%	72.5%	73.75%	–
ESL	50	48%	62%	66%	74%

Table 5: Results of The TOEFL and ESL Questions

Figure 3 is taken from ESL exam [50].

**Text:** A **rusty** nail is not as strong as a clean, new one.

**Word:** Rusty

a) Corroded

b) Black

c) Dirty

d) Painted

**Solution:** (a) Corroded

Figure 3: Sample of ESL exam

Figure 4 is taken from TOEFL exam [50].

**Word:** Levied **Choices:**

a) Imposed

b) Believed

c) Requested

d) Correlated

**Solution:** (a) Imposed

Figure 4: Sample of Toefl exam

## CHAPTER 3

### METHODS

This chapter contains the process of measurement of semantic relatedness and they will explain clearly. Then, information is given about morphological parser and disambiguator. Some morphological parser and disambiguator tools will be explained.

#### 3.1 Measurements of Semantic Relatedness

In this thesis, Semantic Relatedness (SR) is measured for word pairs in Turkish. The number of studies is not adequate on SR for Turkish. This thesis is a new study for Turkish and the thesis proposes a new perspective for Turkish and other agglutinative languages to measure SR. In Turkish language, sufficiently large corpus is difficult to find, so Wikipedia is identified as corpus. Wikipedia was taken at 11.10.2013 to be used in the study. Wikipedia contains a lot of information about different topics, so the number of unique words is high and this situation might affect SR results positively. This thesis focuses on the effect of the morphological structure on SR for Turkish. While SR is being calculated, LSA method is used in the thesis.

##### 3.1.1 Parsing Corpus

In this study, Semantic Relatedness is calculated using the LSA method. Wikipedia provides all data to the users in one big eXtensible Markup Language (XML) format. This XML file should be parsed conveniently to avoid data loss. XML file contains 225519 articles available in Turkish Wikipedia.

While the corpus is being parsed, redundant data are removed from corpus such as metadata, XML tags, image links, hyperlinks and etc. This step is important for

next steps because if any redundant data is skipped without removing to mislead the measurement of SR as the noise in the data will be high in the co-occurrence matrix.

### 3.1.2 Building Dictionary

Before matrix is created from parsed data, all data should be added to the dictionary. While data is being added to the dictionary, each word is read one by one and if the word contains any special characters such as commas, semicolons and dots, this character is removed from the word. Adding operation to the dictionary operation is performed in 6 different ways to examine the effect of Turkish Morphology on SR. These ways are as follows:

- Each word is added to the dictionary without any changes on the morphology of the word.
- Another one is that the root of each word is decided according to the meaning of the word in the sentence using the Sak Morphological Disambiguator tool [46], then the root is added to the dictionary.
- To obtain the effect of first derivational affix, root and all inflections up to the first derivational affix inclusively for a word are added to the dictionary. Sak tool [46] is used for the morphological operation in this adding process to the dictionary.
- To obtain the effect of last derivational affix, root and all inflections up to the last derivational affix inclusively for a word are added to the dictionary. Sak tool [46] is used for the morphological operation in this adding process to the dictionary.
- The root of each word is added to the dictionary. The root of a word is decided using Zemberek. Zemberek lists up all morphological possibilities according to the popularity of a word. In this thesis, the most popular state of a root for a word is used to add to dictionary.
- Finally, Snowball Stemmer tool is used to detect the root of a word. Detected roots are added to the dictionary.

Adding Type	Number of Unique Term
No Stemmer	1,725,285
Root (Hasim Sak)	1,334,646
First Derivational Affix	1,294,739
Last Derivational Affix	1,352,469
Zemberek Stemmer	1,230,319
Snowball Stemmer	1,035,541

Table 6: The Number of Unique Terms

Table above shows the information about the dictionary with different morphological states.

Morphological Disambiguator tool, Zemberek and Snowball will be explained in detail in later sections.

### 3.1.3 Building Co-occurrence Matrix

After building dictionary from the corpus successfully, co-occurrence matrix should be created in order to apply LSA method. The number of rows and columns of co-occurrence matrix is equal to the number of word in the dictionary. For this reason it is square matrix. All words in the matrix are converted to lower-case and all punctuation marks are removed. Sliding window is built and window size is decided. Sliding window contains words depending on the window size. For example in Figure 5, our window size is 1 and we check that how many times *suffered* and *chopin* and also *suffered* and *from* occur together and then increment their co-occurrence value. This operation is repeated for each word in the corpus.

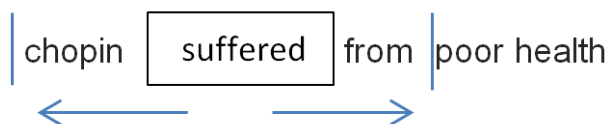


Figure 5: The sample of WordNet

Six different matrices are created using six different dictionaries with six different morphological processing methods described in Section 3.1.2. While the matrix is being created, word frequency and window size constraints are applied to the

corpus.

Frequency refers to how many times a word occurs in the corpora. Applied frequency filters ignore low frequency words as their co-occurrence row will not have many values. Window size and word frequency affect the results directly thus we performed experiments with 5 different window sizes (1, 2, 3, 5 and 10) and 4 different frequencies (10, 20, 30 and 40). Table 7 shows the number of unique terms for each dictionary after word frequency filters are performed on the dictionary.

	10	20	30	40
No Stemmer	216,314	137,927	105,344	86,805
Root (Hasim Sak)	114,058	70,368	53,482	44,355
First Derivational Affix	129,693	81,752	62,639	52,042
Last Derivational Affix	139,040	87,674	67,009	55,538
Zemberek Stemmer	115,476	71,496	54,231	44,956
Snowball Stemmer	106,262	66,597	50,716	42,112

Table 7: The Number of Unique Terms After Frequency Filter

Rows of Table 7 represent morphological form type, and columns represent the value of word frequency.

### 3.1.4 Weighing the Co-occurrence Matrix

Common words are words that appear frequently in texts. For example “a”, “an” and “the” are some known common words in English. The number of common words is high in co-occurrence matrix. The probability of common words’ being used together with any other word is high, this situation might mislead the results. For this reason, entropy is applied to weighed co-occurrence matrix to decrease the effects of common words. To apply the entropy, probability of the word should be calculated . Equation below calculates the probability of co-occurrence of two words  $(w_i, w_j)$  in the corpora.

$$p(w_i, w_j) = \frac{C_{ij}}{N} \quad (3.1)$$

where,  $N$  refers to the number of words in the corpora and  $C$  represents the co-occurrence matrix.

While the weighted matrix  $A$  is calculated, calculated probability  $p(w_i, w_j)$  multiplies the entropy of the second term. Entropy is applied to the matrix using Equation 3.2 [33]:

$$A_{ij} = \log(1 + C_{ij}) \left( - \sum_k p(w_i, w_k) \log(p(w_i, w_k)) \right) \quad (3.2)$$

### 3.1.5 Dimension Reduction Using SVD

SVD is an important step for LSA. SVD can be applied to any matrix and it is decomposed into three matrices as in the equation below:

$$A_{m \times n} = U_{m \times m} S_{m \times n} V_{n \times n}^T \quad (3.3)$$

In the equation above,  $U$  and  $V$  are orthogonal matrices, which means  $U^T = U^{-1}$  and  $V^T = V^{-1}$ , and  $S$  is a diagonal matrix which has zeros in its non-diagonal items.

In this study, created weighed matrix is truncated to  $k$  dimensions using singular values. The  $k$  is a parameter in truncated SVD. Truncated SVD reduces the size of matrix and increases the signal to noise ratio of the matrix.

There is no rules or formulas, while the value of  $k$  is being identified. It may change depending on the size of corpus. In the Ercan study [11],  $k$  is identified as 400 for Turkish Wikipedia. For the English language, when the value of  $k$  is 700, the highest score is obtained in Ercan study. In this thesis, three different  $k$  values (200, 400 and 600) are tested and these are compared with each other. When SVD is applied to matrices,  $k$  value is accepted as 400, but only when the effect of first derivational affix is examined, also different  $k$  values (200 and 600) are tested.

SVDLIBC <sup>1</sup> is an accurate implementation of SVD. It was developed using C libraries [44]. In this thesis, all matrices are decomposed using SVDLIBC which support sparse and dense matrix in the form of text and binary format.

---

<sup>1</sup>It is taken from <http://tedlab.mit.edu/~dr/SVDLIBC/>



## 3.2 Morphological Parser and Disambiguator

### 3.2.1 Morphological Parser

A word contains a root and affixes; in other words, one root and multiple affixes. In the agglutinative languages, vast amount of suffixes can be added to the end of the words. For the Turkish language, the most known sample is “uygarlaştıramadıklarımızdanmışsınızcasına” [37]. The word can be divided into the root and suffixes as follows:

uygar + laş + tır + ama + dık +lar + ımız + dan + mış + sınız +casına

In English, it can be translated as follows:

as if you are among those whom we could not cause to become civilized

As it can be seen in the example above, a lot of new words can be derived from a starting word. The word is divided into a root and affixes for the agglutinative languages. Some agglutinative languages are Turkish, Hungarian, Sumerian, Finnish and Basque.

The process of decomposing words to their morphological structure is called morphological parsing. The difference between morphological parser and disambiguator is that parser lists all candidate parsing forms and disambiguator select one of them according to the meaning of the word in the sentence. This thesis examines different treatment strategies for suffixes in Turkish language. To do this, three different morphological parsers are used.

#### 3.2.1.1 Snowball

Snowball<sup>2</sup> is another stemming algorithm for Turkish language. In addition, Snowball provides a stemmer for other languages such as Romance (English, Spanish, Portuguese, Italian, Romanian), Germanic (German, Dutch), Scandinavian (Swedish, Norwegian, Danish). Snowball is a general stemmer which can be used to develop a stemmer for different language. Snowball for Turkish [14, 13] aims to find the noun stems in a word. Turkish morphological structure is modelled in a Finite State Machine (FSM). Snowball is developed using different programming

---

<sup>2</sup> It can be taken from <http://snowball.tartarus.org/>

languages for different languages. Table 8 shows information about some of them<sup>3</sup>  
:

Stemmer	Programming Language	Received
Russian	php5	11/2005
English	ANSI C	01/2006
German	python	05/2006
Turkish	java	12/2006
English	C#	04/2007
Italian	C#	08/2008
Portuguese	java	11/2008

Table 8: Other Language Snowball Stemmer

Some examples, which are taken from Çilden E. study [14], are shown in Table 9.

Word	Morphological Analysis	Root
Kalelerimizdekilerden	Kale-lAr-UmUz-Da-ki-lAr-Dan	Kale
Çocuğuymuşumcasına	Çocuk-(s)U-(y)mUş -(y)Um-cAsInA	Çocuk
Kedileriyle	Kedi-lAr-(s)U -(y)lA	Kedi
Çocuklarımış	çocuk-lAr-(s)U-(y)lA	Çocuk
Kitabımızada	kitap-UmUz-(y)DU	Kitap

Table 9: Turkish Snowball Examples [14]

Snowball does not contain any lexicon so it can produce invalid words but it is faster than Zemberek and Hasim Sak morphological parser because they check the correctness of produced word using lexicon.

In this thesis, Snowball for Turkish is used to detect the root of each word in Turkish Wikipedia. Detected roots are added to the dictionary, which is used to measure SR.

### 3.2.1.2 Zemberek

Zemberek [1] is developed to fill a gap in the NLP for Turkish. It supports not only Turkish, but also other Turkic languages. It is an open source application. It

<sup>3</sup> It can be taken from <http://snowball.tartarus.org/otherlangs/index.html>

provides some NLP operations such as word suggestion, stemming, spell checking, word construction and **morphological parser** to the researchers. This thesis uses the morphological parser feature of Zemberek.

Word is analyzed through morphological parser and there are four steps in this analysis process which are as follows:

- First, the word is prepared for the parsing operation. All characters are converted to lower case.
- Parts of the word that can also be a root itself are identified.
- For each identified root, suffixes are added to the root. At the end of the process, if the input word and the created word matches each other, input word can be parsed as the created word.
- Resulting word formations are returned.

For each word, the root of the most popular candidate word formation in the results provided by Zemberek is added to the dictionary. If the word can not be parsed by Zemberek, this word is added to dictionary without any changes.

### 3.2.1.3 Morphological Parser

Morphological Analyzer or Parser is a step prior to MD which identifies the morphological structure of words. Additionally, MD tool of Sak et. al. use an MP tool which is developed by Sak et. al [47]. This MP tool was developed using Python programming language.

This MP consists of three components which are lexicon listing, morphotactics and morphophonemics components. These can be described as follows respectively:

- Lexicon Listing Component contains the root of the words.
- Morphotactics Component can be described as suffixes that can be added to the words with an ordered formation.
- Morphophonemics Component is the phonological difference that occurs after adding suffixes to the word.

Hasim Sak morphological parser study is rule based method. Finite State Transducers (FST)[25] are created using the indicated component above. In the input side of the transducer, morphological features are identified for the word and in the output side of the transducer, phonological rules are applied to the word.

Table 10 shows all the possible parses of all words in a sentence (“Bütün insanlar hür , haysiyet ve haklar bakımından eşit doğarlar .”)<sup>4</sup>.

### 3.2.2 Morphological Disambiguator

The main difference between morphological parser (MP) and morphological disambiguator (MD) is that MD decides the root and stems of a word according to the meaning of the word in the sentence. Using this method, the correct forms of the words in the corpus can be added to the dictionary and results are expected to be more reliable than the morphological parser. Certainly, the reliability of the results depends on the accuracy of the used disambiguator tool. In this thesis, an MD tool is used to examine the effect of the root and stems on the similarity of the words. Using the words output of MD are added to the dictionary in three different ways (Root, First Derivational Affix, Last Derivational Affix) which are explained in Section 3.1.2. Many MD studies have been developed for the Turkish language [46, 16, 17, 36, 63], some of which are explained in the following sections.

#### 3.2.2.1 Yüret and Türe Morphological Disambiguator

As it is known, MD is an essential problem for agglutinative languages such as Turkish, Finnish, Basque and etc. Yüret and Türe study [63] is a rule based method to decide on the root and stems of a word according to the meaning of the word in a sentence. In the Turkish language, there are a lot of ambiguous words. For example, Table 11 shows the “masalı” word in different meanings:

In Table 11, “masalı” word has two different roots. First two roots are used with the meaning of “fable” and the last one is used with the meaning of “table”. The main difference between MP and MD is that MD can detect the morphological structure of a word in the sentences. The Yüret D. method provides MD algorithm. Table 11 contains some tags taken from Oflazer et. al. study [35] involving

---

<sup>4</sup> It can be taken from <http://tools.nlp.itu.edu.tr/MorphAnalyzer>

Word	Morphological Parser Results
Bütün	Bütün[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom] : 15.1318359375 bütün[Adj] : 7.4609375
insanlar	bütün[Noun]+[A3sg]+[Pnon]+[Nom] : 12.7197265625 insan[Adj]-[Noun]+lAr[A3pl]+[Pnon]+[Nom] : 20.8125 insan[Noun]+lAr[A3pl]+[Pnon]+[Nom] : 10.1357421875
hür	hür[Adv] : 16.125 Hür[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom] : 16.517578125 hür[Adj] : 11.40234375
,	,[Punc] : 16.125
haysiyet	haysiyet[Noun]+[A3sg]+[Pnon]+[Nom] : 13.03515625
ve	ve[Conj] : 3.8681640625
haklar	hâk[Noun]+[NoHats]+lAr[A3pl]+[Pnon]+[Nom] : 10.6025390625 hakla[Verb]+[Pos]+Hr[Aor]+[A3sg] : 19.09375 hakla[Verb]+[Pos]+Hr[Aor]+[A3sg] : 19.5458984375 Hak[Noun]+lAr[A3pl]+[Pnon]+[Nom] : 10.4658203125 hak(I) [Noun]+lAr[A3pl]+[Pnon]+[Nom] : 19.65625 hak[Adj]-[Noun]+lAr[A3pl]+[Pnon]+[Nom] : 21.9111328125
bakımından	bakım[Noun]+[A3sg]+SH[P3sg]+NDAn[Abl] : 13.083984375 bakım[Noun]+[A3sg]+Hn[P2sg]+NDAn[Abl] : 17.7568359375 bakımından[Adv] : 16.125
eşit	eşit[Adj] : 8.9111328125 Eşit[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom] : 12.26953125
doğarlar	doğ[Verb]+[Pos]+Ar[Aor]+lAr[A3pl] : 16.23828125 doğ[Verb]+[Pos]+Ar[Aor]+lAr[A3pl] : 16.6103515625
.	.[Punc] : 16.125

Table 10: Morphological Parser Results

<b>masal</b> + Noun+A3sg+Pnon+Acc
<b>masal</b> + Noun+A3sg+P3sg+Nom
<b>masa</b> + Noun+A3sg+Pnon+Non^DB+Adj+With

Table 11: Parses of “Masah” [63]

126 tags. In this study, subsets are created from words which contain each unique tag. Then, by analyzing the subsets using Greedy Prepend Algorithm (GPA) the rules are acquired and decision lists are formed. After the rules have been learnt in order to predict the morphological structures of words, firstly morphological analyzer lists all possible forms of the words. Then, decision lists are used to predict the parse of the words. The achievement rate of this method is stated as %96 using Turkish news (totally one million words).

### 3.2.2.2 Hasim Sak Method

This thesis is used Sak et. al. MD tool [46]. This tool consists of two parts which are morphological parser and morphological disambiguator. Morphological parser is explained in Section 3.2.1

#### Morphological Disambiguator

In this MD study [46], the probability of all candidate parses are decided using Hakkani - Tür D.’s trigram based model [17]. Then Viterbi algorithm is used to decode the n-best candidates from the calculated parse probability. Subsequently, Perceptron algorithm is applied to order the candidate parses.

The MD tool was developed using Perl programming language. When the MD tool is used, the tool needs candidate parses of each words of a sentence. The following sample disambiguation is processed according to candidate parses which are created in 3.2.1.3 and disambiguation results are shown in Table 12 :

Word	Disambiguation Results
Bütün	bütün[Adj] Bütün[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom]
insanlar	insan[Noun]+lAr[A3pl]+[Pnon]+[Nom]
hür	hür[Adj]
,	,[Punc]
haysiyet	haysiyet[Noun]+[A3sg]+[Pnon]+[Nom]
ve	ve[Conj]
haklar	hâk[Noun]+[NoHats]+lAr[A3pl]+[Pnon]+[Nom]
bakımından	bakım[Noun]+[A3sg]+SH[P3sg]+NDAn[Abl]
eşit	eşit[Adj]
doğarlar	doğ[Verb]+[Pos]+Ar[Aor]+lAr[A3pl]
.	.[Punc]

Table 12: Disambiguation Results

## CHAPTER 4

### EXPERIMENTS and RESULTS

This thesis extends the studies of Bullinaria [7] and Ercan [11]. In the Bullinaria study, the effect of stems are evaluated using different parameters for English language. In Turkish, the first study about semantic relatedness is developed by Ercan[11]. We extends the work on Turkish language by investigating the effects of the morphology is analysed on semantic relatedness in Turkish.

In this chapter, we test the effects of morphology on Semantic Relatedness for Turkish language performing some experiments.

In this chapter, firstly all experiments are explained and then the results of all experiments are given.

#### 4.1 Experiments

##### 4.1.1 Clustering Purity

In this experiment, words in the word set are categorized according to the results of semantic relatedness. The word set are taken from Mitchell et. al. [32] study. When the word set is categorized, 12 different categories are given to nine people and also is identified that each category consists of 5 words. 60 different pictures of words are shown to the people six times and people assign these words to the categories. For example; two categories are shown in Table 13. However the word set is prepared in English so the word set is translated in Turkish.

When the word set is translated to Turkish, three words(desk, igloo, arch) are removed from word list, as both “desk” and “table” is used in the same category, which can both be translated to the same word “masa” in Turkish .In addition, “arch” meaning is used in the clothes category in Turkish but in this word set, it



<b>Animals</b>	<b>Plants</b>
bear	carrot
cat	celery
cock	corn
dog	lettuce
horse	tomato

Table 13: Two Sample Categories

is used in the building parts. Finally, the frequency of “igloo” is low in Turkish Wikipedia thus it is not added to the dictionary. Each word in the data set is compared with each other and the categorization is performed according to SR result. CLUTO Clustering Toolkit [27] is used to categorize the word list. In the CLUTO Clustering Toolkit, k way clustering algorithm [8] is applied to the word set.

Using SR functions, words in the word set are clustered. These clusters are compared with people’s categories and depending on the number of matches, clustering purity is calculated, using Equation 4.1:

$$ClusteringPurity = \frac{positive}{positive + negative} \quad (4.1)$$

where, positive refers to the number of correctly clustered word and negative refers to the number of incorrectly clustered word.

#### 4.1.2 Word Association

Table 14 shows different methods to be investigate the corpora to evaluate SR.

To perform these identified experiments, a list of word pairs is used. The list consists of 101 word pairs. The word pairs are given a score according to their semantic relatedness with each other by 44 people. People rate the word pairs from 1 to 5. The average is calculated according to the given scores. SR is calculated on the same word pairs using the algorithm developed for this thesis. Correlation is calculated between the calculated value and the average of the people’s rating. Pearson Correlation [52] is used to calculate the correlation. As

Test Conditions
No Stemmer Results
Three Different Morphological Parser Results
Morphological Disambiguator Results
Windows Size
Different SVD truncate values

Table 14: Information about to be applied tests

shown in Equation 4.2.

$$r = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{\sum X^2 - \frac{(\sum X)^2}{N}} \sqrt{\sum Y^2 - \frac{(\sum Y)^2}{N}}} \quad (4.2)$$

Where , X is the average of people rating, Y is the program score for the word pairs, N is the number of word pairs and r is the result of the correlation.

After the experiments are performed, T-test [40] is performed between the experiments in order to test if the correlations are significantly different. When the T-test is applied between two correlation values, Equation 4.3 is used:

$$t = (r_{jk} - r_{jh}) \sqrt{\frac{(n-1)(1+r_{kh})}{2((n-1)/(n-3)|R| + \bar{r}^2(1-r_{kh})^3)}} \quad (4.3)$$

where,  $r_{jk}$  is the correlation of people score to first algorithm's score,  $r_{jh}$  is the correlation of the people score to second algorithm's score,  $r_{kh}$  is the correlation of the score between the algorithms, n is the number of word pairs,  $R = 1 - r_{jk} - r_{jh} - r_{kh} + (2 * r_{jk} * r_{jh} * r_{kh})$  and  $\bar{r} = \frac{r_{jk} + r_{jh}}{2}$

In the following SR experiments, stated values are the correlation of the word pair list.

## 4.2 Results

### 4.2.1 Effects of Morphology on Semantic Relatedness

#### 4.2.1.1 No-Stemmer Experiment

In the No-Stemmer, when the dictionary is created, words are kept in the dictionary without any changes. For this reason, the number of unique words in the dictionary is very high as 1,725,285. Table 15 shows the results of SR.

		Windows Size				
		1	2	3	5	10
Frequency	10	0.5759	0.5833	0.6080	0.6428	0.6581
	20	0.5795	0.5908	0.6066	0.6393	0.6584
	30	0.5582	0.5706	0.5891	0.6191	0.6352
	40	0.5318	0.5483	0.5611	0.5855	0.6014

Table 15: No-Stemmer Word Association Experiment

Table 15 has two different parameters which are frequency and windows size. Frequency refers to how many times each word appears in the corpora. If the number of occurrences of the word in corpora is less than the identified value, the word is ignored and excluded from the dictionary. In the No-Stemmer experiment, a word can appear in more than one different morphological form in the corpora. In addition, for this reason the number of unique words is larger than other alternatives.

According to Table 15, the achievements of SR increases with the window size which contradicts with experiments in English performed by Bullinaria [7]. The reason of this discrepancy can be size of the used corpus as a larger corpus can provide opportunity to create more reliable vectors for each word. The rate of success is as high as 0.6584874504, when frequency is 20 and window size is 10.

In Figure 6, it is seen that the diamond points (frequency 10) and square points (frequency 20) almost overlap. In addition, the achievement increases in direct proportion with windows size.

Consequently, the success of the No-Stemmer experiment results is not high as

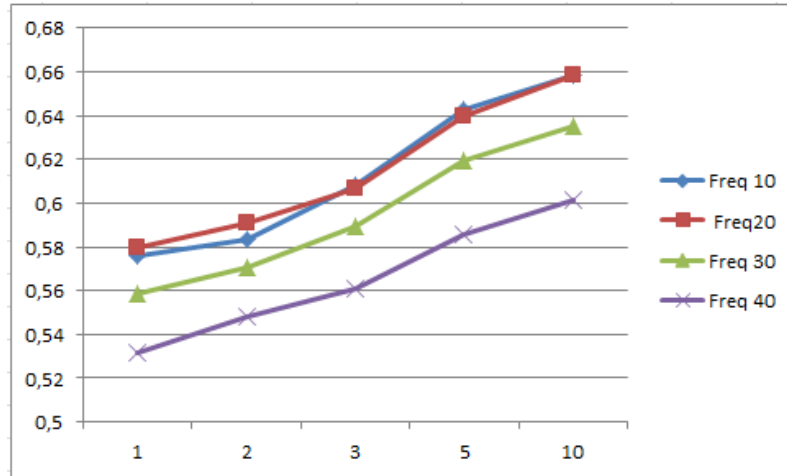


Figure 6: No-Stemmer word association experiments

the result of other experiments in this thesis

Figure 7 shows the result of clustering purity experiment for No-Stemmer experiment. The most successful result is obtained when the frequency is 40 and windows size is 10.

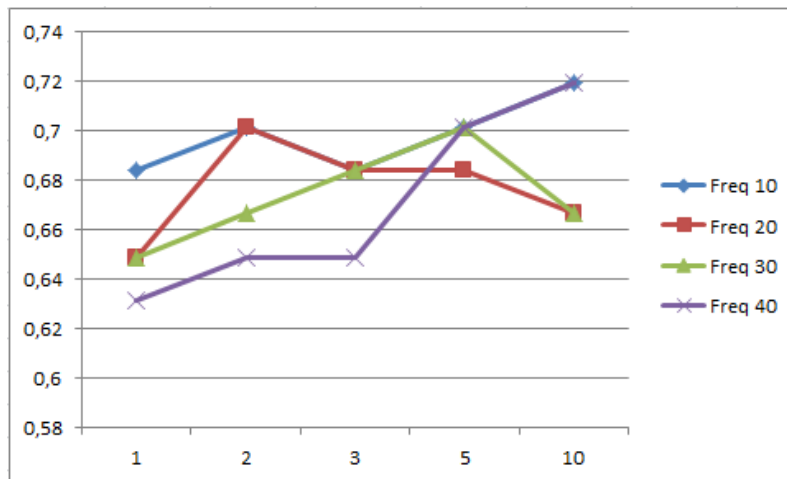


Figure 7: No-Stemmer clustering purity experiments

#### 4.2.1.2 Snowball Experiment

Snowball is the first morphological processor experiment for this thesis. In this experiment, the root of each word in the corpus is detected using a suffix stripping stemmer and added to the dictionary. This situation decreases the number of low frequency words as a word can take multiple inflections suffixes and derivational affixes in Turkish. If the root of the word is added to the dictionary, the frequency

of the root is equal to the total number of words derived from this root. The number of unique words in the dictionary is 1,035,541. The number of unique words in the dictionary created using Snowball Stemmer, is approximately 680,000 less than the number of words in the No-Stemmer dictionary.

Table 16 shows the semantic relatedness results of Snowball experiment. According to Table 16, the highest point is 0.6548707819, where the window size is 2 and frequency is 30.

		Windows Size				
		1	2	3	5	10
Frequency	10	0.6299	0.6380	0.6400	0.6340	0.6283
	20	0.6298	0.6491	0.6404	0.6318	0.6377
	30	0.6317	0.6548	0.6439	0.6358	0.6456
	40	0.6314	0.6547	0.6481	0.6298	0.6390

Table 16: Snowball Stemmer Word Association Experiment

In Figure 8, the results are dispersed not linearly in contrast to the No-Stemmer experiment and the results of this experiment are closer each other. The movements of lines resemble each other. It can be said that keeping the frequency at the level between 30 and 40 and window size at the level of two increase the achievement of SR for the users of Turkish Snowball Stemmer.

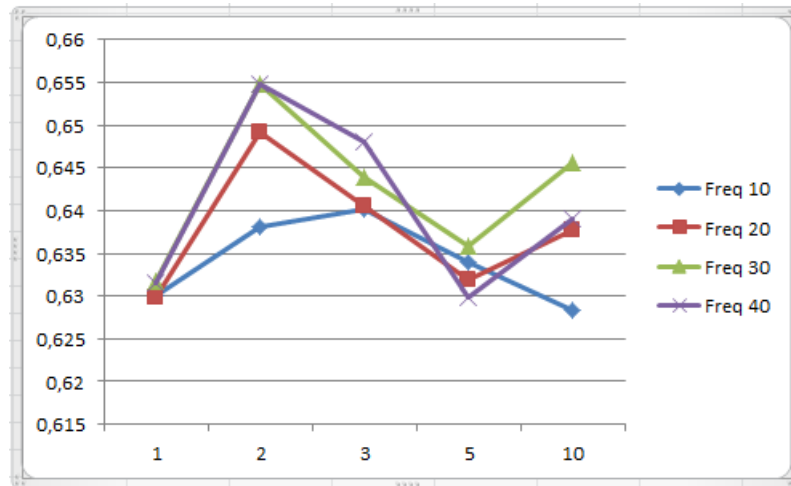


Figure 8: Snowball word association experiments

The result of clustering purity experiment for Snowball is shown in Figure 9. The highest score is obtained when the window size is 5 and word frequency is 20.

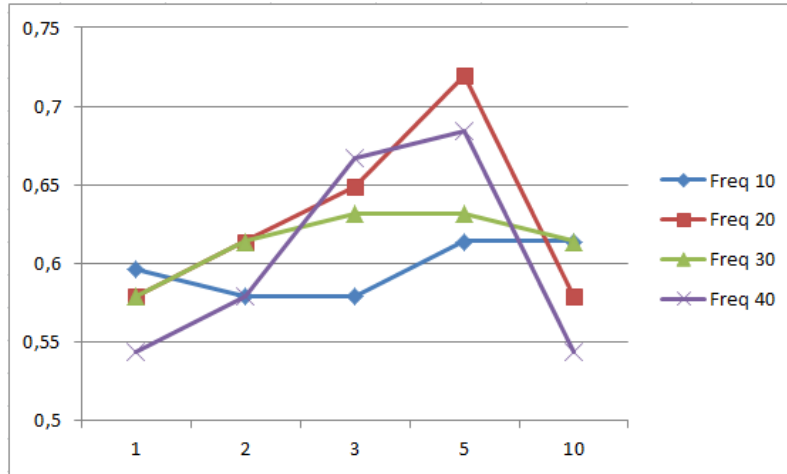


Figure 9: Clustering purity for snowball experiments

#### 4.2.1.3 Zemberek Experiment

In this experiment, words are processed by Zemberek [1] and each word is divided into root and suffixes according to its morphological structure and the word root is added to the dictionary. Zemberek produces all possible morphological forms of a word and it returns them in sorted order according to popularity of the word. The root of the most common parse of the word is added to the dictionary. In the created dictionary, the number of unique words is 1,230,319. The number of unique words in the dictionary created using Zemberek is more than the number of unique words in the dictionary created using Snowball. Difference between Snowball and Zemberek is that while Snowball clips all stems, Zemberek separates words into root and stems.

		Windows Size				
		1	2	3	5	10
Frequency	10	0.7242	0.7604	0.7578	0.7636	0.7422
	20	0.7286	0.7629	0.7593	0.7634	0.7412
	30	0.7319	0.7645	0.7650	0.7689	0.7441
	40	0.7492	0.7658	0.7646	0.7650	0.7459

Table 17: Zemberek Parser Experiment Semantic Relatedness Results

Zemberek experiment results are shown in Table 17. The best score in this experiment is 0.768961581, obtained when the frequency is 30 and windows size is 5.

Figure 10 shows the results in the plot. In the graph, the highest score is clearly observed (triangle points). In addition, changes of the shape of functions are similar to each other. In the all results of Zemberek experiment, it can be seen that if the windows size is greater than 5, the success of the result decreases. When the windows size is 5, achievement of the experiment is the highest for all frequencies in this experiment.

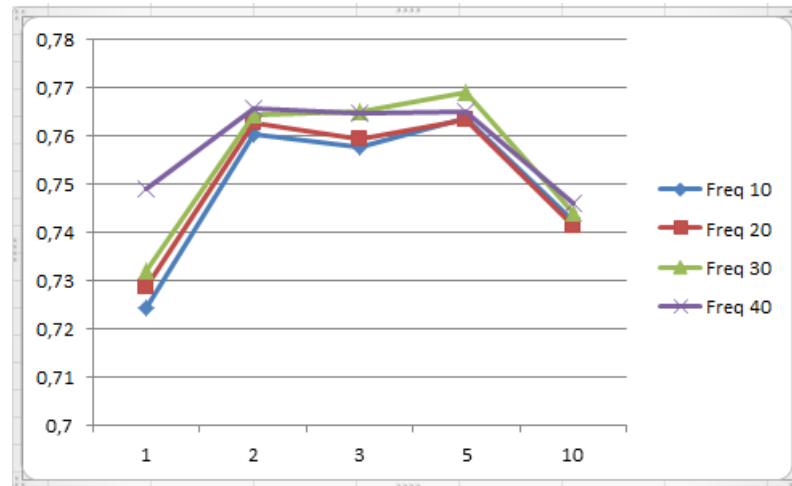


Figure 10: Zemberek word association experiments

T-test is applied between Snowball and Zemberek according to the their correlation coefficient results. The result of the test shows that Zemberek is significantly better than Snowball.

The result of the clustering purity experiment is shown in Figure 11.

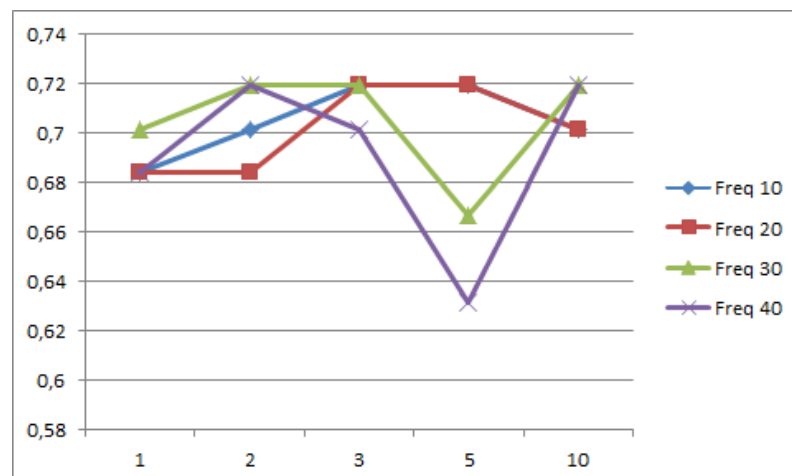


Figure 11: Zemberek clustering purity experiments

The highest score is obtained when window size is 5 and word frequency is 20.

#### 4.2.1.4 Hasim Sak Morphological Parser

In this experiment, Sak parser [47] is used to detect the root of the word. The semantic relatedness is measured according to these roots. While the word is being parsed by Sak parser, there may be more than one different parsing results for a word. Negative log probability [48] is calculated for each parsed result of a word. It is way to learn morphosyntactic rule from corpus. It is the probability of a word and its affixes to appear in the corpus. Minimum log probability state is selected between all parsed states of the word. After all roots are added to the dictionary, total number of unique word is 1,336,269 in the dictionary . The total number of unique words in the dictionary, which is produced by the Sak parser, is greater than the number of unique words in Zemberek and Snowball dictionary. Sak parser results are indicated in Table 18.

		<b>Windows Size</b>				
		<b>1</b>	<b>2</b>	<b>3</b>	<b>5</b>	<b>10</b>
<b>Frequency</b>	<b>10</b>	0.7217	0.7482	0.74380	0.7693	0.7447
	<b>20</b>	0.7232	0.7547	0.7400	0.7699	0.7519
	<b>30</b>	0.7341	0.7643	0.7468	0.7652	0.7419
	<b>40</b>	0.7377	0.7671	0.7488	0.7667	0.7339

Table 18: Hasim Sak Parser Word Association Experiment

In this experiment, the highest score can be obtained as 0.7699982548 when the windows size is 5 and frequency is 20. When the window size is 5, all scores are greater than 0.765. Figure 12 shows the Sak parser results.

The maximum score in this experiment is the highest score of all the experiments that are performed in this thesis. In addition, the score is seen as higher than all the obtained results from study about the Turkish SR.

The result of clustering purity for Hasim Sak Parser is shown on Figure 13. While the highest score is obtained, when the window size is 5 and word frequency is 30, results are close to each other.

The result of Sak morphological parser is higher than the result of Zemberek but there is significant no difference between them.



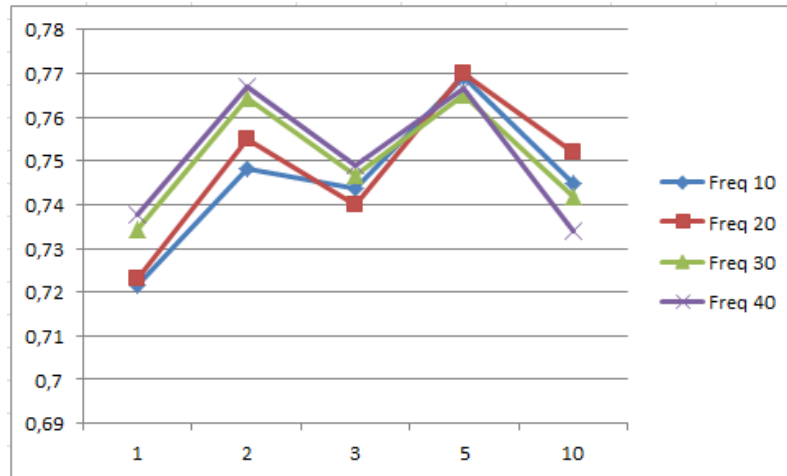


Figure 12: Hasim Sak parser word association experiments

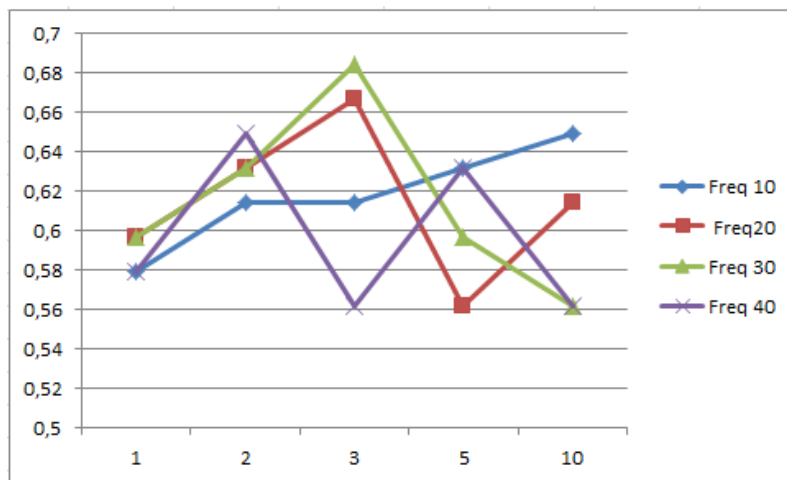


Figure 13: Hasim Sak parser clustering purity experiments

#### 4.2.1.5 Hasim Sak Morphological Disambiguator

Morphological disambiguator determines which meaning of the word is used in the sentence. Therefore, correct parsed form of the word can be added to the dictionary.

Three different experiments are performed using the Sak morphological disambiguator. First, the root is detected by MD and these roots are added to the dictionary. The result of the first experiments is shown in Table 19.

The target of this experiment monitors the change on SR according to the Sak parser, when the word is added to the dictionary using MD. The results obtained will be compared to the result of Sak parser (Section 4.2.1.4). As can be seen

		Window Size				
		1	2	3	5	10
Frequency	10	0.7164	0.7378	0.7279	0.7461	0.7304
	20	0.7156	0.7377	0.7281	0.7472	0.7332
	30	0.7208	0.7390	0.7276	0.7489	0.7289
	40	0.7262	0.7398	0.7284	0.7538	0.7287

Table 19: Hasim Sak Disambiguator Word Association Experiment

from Table 19 the highest score is 0.7538723041 encountered when the window size is 5 and frequency is 40. For all frequencies the highest scores are obtained, when the window size is 5. Figure 14 shows the results for MD.

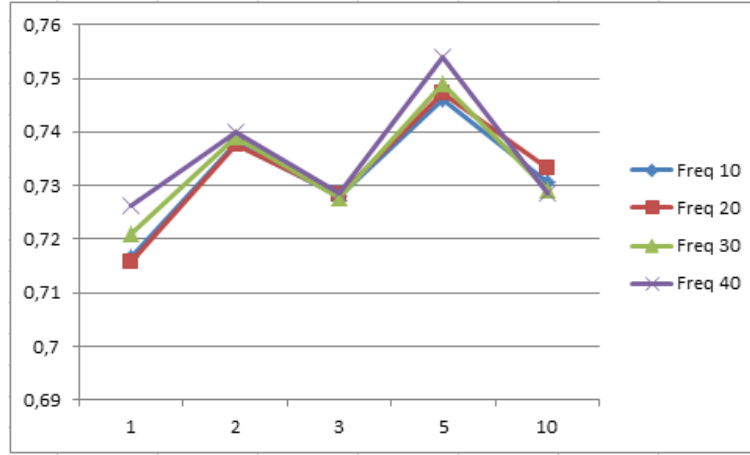


Figure 14: Hasim Sak disambiguator word association experiments

Results of the clustering purity is shown in Figure 15. The results of clustering purity experiment for disambiguator do not move parallel for all frequencies.

The second experiment of the MD detects the effect of the first derivational affixes. For this reason, the word is added to the dictionary as root and all suffixes up to the first derivational affixes inclusively. Derivational affixes changes the meaning of the word, so a lot of new words are added to the dictionary. According to the example below (ölümsüzleştiriveremeyebileceklerimizdenmişsinizcesine) [49]:

ölüm[Noun] – sHz[Adj+Without] – lAş[Verb+Become] – DHr[Verb+Caus]+[Pos]  
–YHver[Verb+Hastily] + YAmA[Able+Neg] – YAbil[Verb+Able] –YAcAk[Noun+FutPart]  
+ lAr[A3pl] + HmHz[P1p1] + NDAn[Abl] – YmHş[Verb+Narr] + sHnHz[A2pl] –  
CAsHnA[Adv+AsIf]

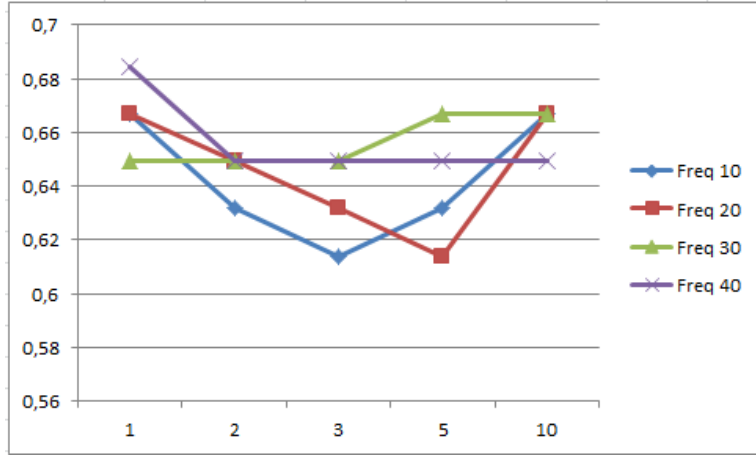


Figure 15: Hasim Sak disambiguator clustering purity experiments

the word is added to the dictionary as *ölümsHz*. In the example minus(-) refers to the derivational affix, plus(+) refers to the inflectional suffix. The word is parsed using the Sak parser. When the word is added to the dictionary, all characters are converted to lower-case.

Table 20 shows this experiment results. The highest score in this table is 0.719221611. The score is less than the score obtained from Zemberek, Sak parser and Sak disambiguator for root. For each frequency, the highest is observed, when the window size is 5. Figure 16 below shows the result of this experiment.

		Window Size				
		1	2	3	5	10
Frequency	10	0.6788	0.7040	0.6949	0.7082	0.6945
	20	0.6786	0.7053	0.7030	0.7144	0.6890
	30	0.6785	0.7063	0.7049	0.7192	0.6847
	40	0.6777	0.6966	0.6987	0.7071	0.6809

Table 20: Disambiguator Experiment For First Derivational Affix

Clustering purity experiment is performed using this morphological form. The result of clustering purity is shown in Figure 17

The last experiment of the MD examines the effects of last derivational affix. To do this, each word in the corpus is added to the dictionary as root and all derivational affixes and inflectional suffixes up to the last derivational affix inclusively. The example [49] below explains this situation clearly:

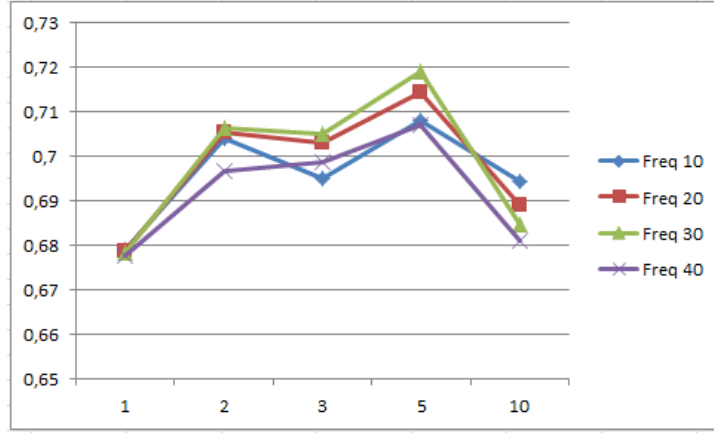


Figure 16: Hasim Sak disambiguator experiments for first derivational affix

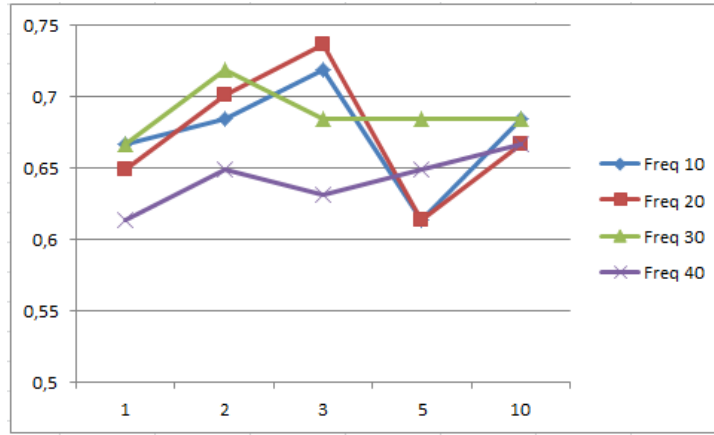


Figure 17: Clustering purity experiments for first derivational affix

ölüm[Noun] – sHz[Adj+Without] – lAş[Verb+Become] – DHr[Verb+Caus]+[Pos]  
 –YHver[Verb+Hastily] + YAmA[Able+Neg] – YAbil[Verb+Able] –YAcAk[Noun+FutPart]  
 + lAr[A3pl] + HmHz[P1p1]

the word is added to the dictionary as ölümSHzLaşDHrYHverYAmAYAbilYAcAk. The word is parsed using Sak parser. When the word is added to the dictionary, all characters are converted to the lower-case. The results of these experiments are shown in Table 21.

According to Table 21 when the window size is 5, correlation values are higher than 0.7 for all frequencies. The highest score is 0.71499734744 in the table. The score is obtained when frequency and window size are 30 and 5 respectively. The lowest score is 0.6735216416 obtained when the frequency and window size are 10, 1 respectively.

The scores are represented in Figure 18. When the window size changes to 5 from

		Window Size				
		1	2	3	5	10
Frequency	10	0.6735	0.6941	0.6980	0.7075	0.6749
	20	0.6808	0.6925	0.6927	0.7104	0.6825
	30	0.6824	0.6909	0.6944	0.7149	0.6771
	40	0.6802	0.6898	0.6912	0.7110	0.6749

Table 21: Disambiguator Experiment For Last Derivational Affix

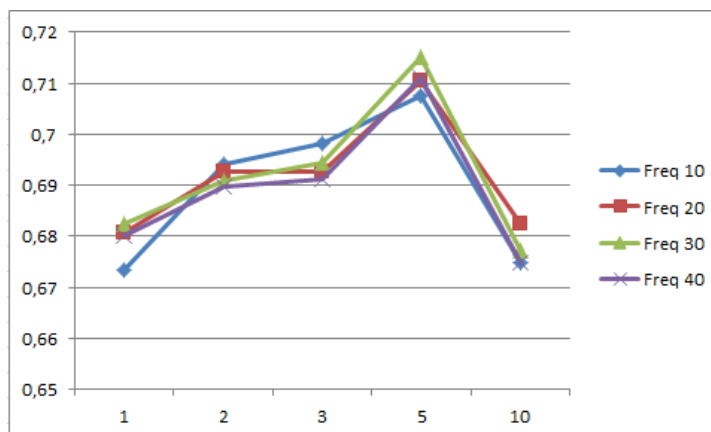


Figure 18: Hasim Sak disambiguator experiments for last derivational affix

3, scores increase rapidly in the figure.

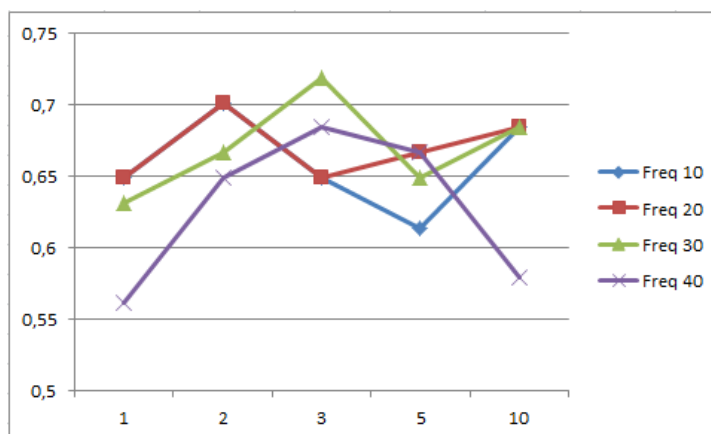


Figure 19: Clustering purity experiments for last derivational affix

Finally, clustering purity experiment is performed for this morphological form. Figure 19 shows the result of clustering purity. In the figure, it can be seen as the results are close to each other for all window sizes.

According to the all word association experiments, the results of morphological

parser experiments (Sak parser and Zemberek) is significantly better than the results of stemmer experiment(Snowball). In addition, the highest score is obtained using Sak morphological parser experiments but there is no significant difference between Zemberek and Sak parser. Also it can be seen that morphological disambiguator does not increase the success of SR.

According to the Clustering purity experiments, Zemberek obtains the most successful results. Zemberek is significantly better than all other experiments. In addition, Sak morphological disambiguator is significantly better than Sak morphological parser, Snowball and No Stemmer experiments. Whereas first derivational affix experiment is significantly indifferent from last derivational affix, it is significantly better than Sak parser.

#### 4.2.2 The Effect of Dimension Reduction

Until this experiment,  $k$  value for SVD is selected as 400 in all experiments. It is decided according to the Ercan study [11], where the highest scores are obtained when the  $k$  value is 400.

In this experiment, two different  $k$  values, which are 200 and 600, are examined. Thus, the effect of the truncated value can be obtained. Noise words can be cleaned using SVD(Singular Value Decomposition). This experiment is performed on only one morphological form which is using root forms of MD. Table 22 values are measured when  $k$  is equal to 200. Results of these experiments are shown in Table 22.

		<b>Window Size</b>				
		<b>1</b>	<b>2</b>	<b>3</b>	<b>5</b>	<b>10</b>
<b>Frequency</b>	<b>10</b>	0.6958	0.7288	0.7366	0.7458	0.7199
	<b>20</b>	0.6999	0.7316	0.7398	0.7503	0.7277
	<b>30</b>	0.6992	0.7338	0.7394	0.7538	0.7261
	<b>40</b>	0.7041	0.7324	0.7380	0.7533	0.7250

Table 22: SVD Truncate Value is 200 for MD

In Table 22, the highest score can be obtained as 0.7538369571. The highest score of the first MD experiment, where the  $k$  is 400, is greater than this experiment score. In addition, when the window size 1, 2 and 10, for all result, first MD

experiment is more successful than this experiment but when the window size is 3, 200 for the truncated  $k$  value is more successful.

		Window Size				
		1	2	3	5	10
Frequency	10	0.7255	0.7336	0.7274	0.7454	0.7274
	20	0.7244	0.7356	0.7346	0.7494	0.7246
	30	0.7242	0.7383	0.7390	0.7523	0.7272
	40	0.7279	0.7409	0.7373	0.75408	0.7278

Table 23: SVD Truncate Value is 600 for MD

Another experiment is performed when the truncated  $k$  value is 600. The result of this experiment is shown in Table 23. The highest score of the experiment is 0.7540828837. The highest score of the experiment is greater than other MD experiments where truncated  $k$  values are 200 and 400.

It can not be said that truncated  $k$  value should be any number because there is a situation where the score is high of each experiment.

The results of the two experiments where  $k$  is 200 and 600, are shown in Figure 20 and Figure 21.

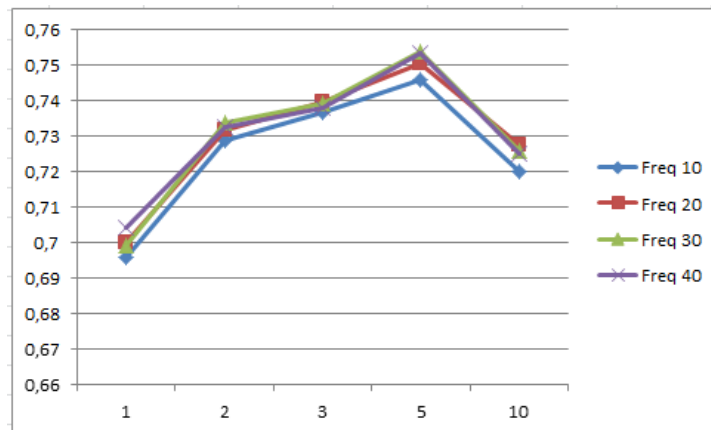


Figure 20: SVD truncate value  $k$  is 200

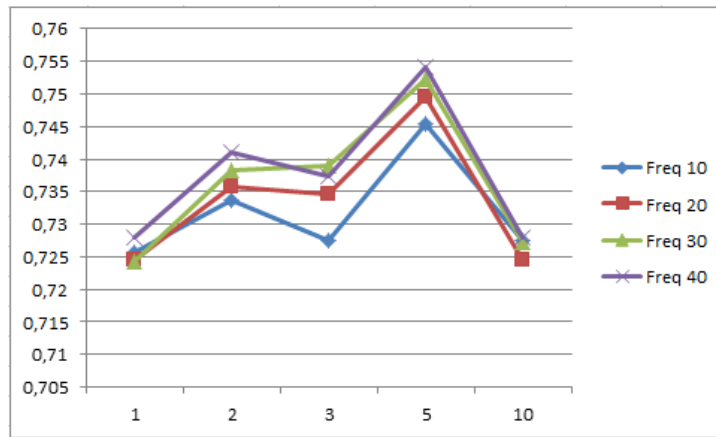


Figure 21: SVD truncate value  $k$  is 600

In Figures 21, it can be seen as changing of the frequency does not effect the result of experiment so much when the  $k$  is equal to 200. On the other hand, when  $k$  is equal to 600, frequency is more effective than previous situation.



## CHAPTER 5

### CONCLUSION AND FUTURE WORK

Different processing techniques are evaluated for Turkish semantic relatedness calculations but this thesis focuses on the effect of the morphology on the semantic relatedness for Turkish. For example, “Ekmek” has two different meanings that are “Bread” and “to Plant” in Turkish. The meaning of the word in the context is decided using morphological disambiguator and the effects of morphological disambiguator are tested on SR. The results of this study light the way of researchers who want to work SR on other agglutinative languages. While the effects of morphology are being tested, three different morphological forms, which are root, first derivational affix and last derivational affix of the words in the corpus, are investigated. In addition, difference between the results of morphological parser and disambiguator is examined and also the effect of SVD is tested on SR. Furthermore as a less computationally expensive alternative Snowball stemmer is used to remove the stems of word and SR is calculated using these clipped word.

The SR methods are performed starting from cheapest to expensive respectively (No-Stemmer, Snowball, Zemberek, Hasim Sak Parser, Hasim Sak Disambiguator). According to the results of our experiments, removing the affixes of words increase the accomplishment of SR on Turkish language. However, morphological processor methods should be used instead of basic methods like Snowball. The most commonly used root of word obtains significantly better results than basic stemmer algorithm but in the Bullinaria [7] study, there is no significant difference between stemmer and lemmatizer for English language. In addition, Morphological Disambiguator experiments are performed on parsed candidates by morphological parser but disambiguator does not increase significantly the success of SR and also the performance of MP is better than MD.

Words have many different forms in Turkish so if the words are added without any changes, same words occur in the dictionary with different forms. This factor affect the result of SR negatively.

In an other experiment, Zemberek tool is used to detect the root of a word and SR is calculated according to these roots. SR result of this study is significantly better than the result of Snowball experiment. In the Zemberek experiment, all result of SR is higher than 72%.

One of the important parts of this thesis is Hasim Sak parser experiment. Using the result of this experiment and morphological disambiguator, we can examine the effects of morphological disambiguator. In this study, the highest score of Hasim Sak parser is seen as higher than all the obtained results from studies about the Turkish SR. When the window size is 5, the result of parser is higher than other window size. All results of this parser are higher than 72%.

When we compare the results of experiments which use morphological parser and morphological disambiguator, we observed that morphological disambiguation does not increase the success of SR for Turkish. According to this experiment, researchers can work with morphological parser instead of morphological disambiguator for SR because morphological disambiguator is computationally more expensive.

The last experiment measures the effects of SVD on SR. When this experiment is performed, three different truncate values are identified for SVD. These values are 200, 400 and 600. The results of the experiment show that there is no significant difference between them so if SVD truncate value is decided between 200 and 600, truncate value should be selected 200 because when cosine similarity is calculated, it is three times more efficient than using 600 dimensions.

## **5.1 Future Work**

While our experiments suggest that MD does not increase the effectiveness of semantic relatedness, additionally experiments using other morphological disambiguation tools can increase the success of semantic relatedness. In addition, the effect of morphology can be tested other agglutinative languages such as Hungarian and Finnish. In addition, our study is limited with Wikipedia as a corpus. If a larger corpora is built for Turkish the effects of corpora size can be investigated.

## REFERENCES

1. **Akın A. A., Akın M. D. (2007)**, “*Zemberek, an Open Source NLP Framework for Turkic Languages*”. *Structure*, vol. 10, pp. 1–5.
2. **Antworth E. L. (1994)**, “*Morphological Parsing with a Unification-based Word Grammar*”. In *Proceedings of the North-Texas Natural Language Processing Workshop*. University of Texas, Arlington, pp. 24–32.
3. **Arısoy E., Sak H., Saraçlar M. (2007)**, “*Language Modeling for Automatic Turkish Broadcast News Transcription*”. In *Proc. Interspeech, Antwerp, Belgium*. Citeseer, pp. 2381–2384.
4. “<http://www.dblab.upatras.gr/balkanet/resources.htm>”, (Data Download Date: 2013).
5. **Banerjee S., Pedersen T. (2003)**, “*Extended Gloss Overlaps as a Measure of Semantic Relatedness*”. In *IJCAI*, vol. 3, pp. 805–810.
6. **Barzilay R., Elhadad M. (1997)**, “*Using Lexical Chains for Text Summarization*”. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pp. 10–17.
7. **Bullinaria J. A., Levy J. P. (2012)**, “*Extracting Semantic Representations from Word Co-occurrence Statistics: Stop-lists, Stemming, and SVD*”. *Behavior research methods*, vol. 44, no. 3, pp. 890–907.
8. **Chan P. K., Schlag M. D., Zien J. Y. (1994)**, “*Spectral K-way Ratio-cut Partitioning and Clustering*”. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 13, no. 9, pp. 1088–1096.
9. “<http://www.nytimes.com/2014/02/10/technology/wikipedia-vs-the-small-screen.html>”, (Data Download Date: 2014).
10. **Elberrichi Z., Rahmoun A., Bentaallah M. A. (2008)**, “*Using WordNet for Text Categorization.*” *Int. Arab J. Inf. Technol.*, vol. 5, no. 1, pp. 16–24.

11. **Ercan G. (2012)**, *Gonenc Ercan. Lexical Cohesion Analysis for Topic Segmentation, Summarization and Keyphrase Extraction*. PhD thesis, Bilkent University, Turkey. pp. 44–88.
12. **Eroğlu O., Kardes H., Torun M., (2009)**. “*Unsupervised Segmentation of Words into Morphemes*”. pp. 5–21.
13. **Eryigit G., Adali E. (2004)**, “*An Affix Stripping Morphological Analyzer for Turkish*”. In Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, Innsbruck, Austria, pp. 299–304.
14. “<http://snowball.tartarus.org/algorithms/turkish/accompanying-paper.doc>”, (Data Download Date: 2006).
15. **Gabrilovich E., Markovitch S. (2007)**, “*Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis*.” In IJCAI, vol. 7, pp. 1606–1611.
16. **Görgün O., Yıldız O. T. (2012)**, “*A Novel Approach to Morphological Disambiguation for Turkish*”. In Computer and Information Sciences II. Springer, pp. 77–83.
17. **Hakkani-Tür D. Z., Oflazer K., Tür G. (2002)**, “*Statistical Morphological Disambiguation for Agglutinative Languages*”. Computers and the Humanities, vol. 36, no. 4, pp. 381–410.
18. **Hamp B., Feldweg H. (1997)**, “*GermaNet - a Lexical-Semantic Net for German*”. In In Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications, pp. 9–15.
19. **Hassan S., Mihalcea R. (2011)**, “*Semantic Relatedness Using Salient Semantic Analysis*”. In AAAI, pp. 884–889.
20. **Heemskerk J. S., van Heuven V. et al. (1993)**, “*MORPA, a Morpheme Lexicon Based Morphological Parser*”. In In Analysis and Synthesis of Speech: Strategic Research Towards High- Quality Text-to-Speech Generation. Mouton de Gruyter, pp. 67–85.
21. **Hirst G., St-Onge D. (1998)**, “*Lexical Chains as Representations of Context for The Detection and Correction of Malapropisms*”. WordNet: An electronic lexical database, vol. 305, pp. 305–332.

22. **Hughes T., Ramage D. (2007)**, “*Lexical Semantic Relatedness with Random Graph Walks*”. In Proceedings of The 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 581–589.
23. **Islam A., Inkpen D. (2006)**, “*Second Order Co-occurrence PMI for Determining The Semantic Similarity of Words*”. In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2006), pp. 1033–1038.
24. **Jiang J. J., Conrath D. W. (1997)**, “*Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy*”. CoRR, vol. cmp-lg/9709008, pp. 1–13.
25. **Jurafsky D., Martin J. H. (2000)**, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, pp. 120–223.
26. **Kim S. N., Baldwin T. (2005)**, “*Automatic Interpretation of Noun Compounds Using WordNet Similarity*”. In Natural Language Processing–IJCNLP 2005. Springer, pp. 945–956.
27. “<http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview>”, (Data Download Date: 2014).
28. **Landauer T. K., Dumais S. T. (1997)**, “*A Solution to Plato’s problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge.*” Psychological review, vol. 104, no. 2, pp. 211.
29. **Li X., Szpakowicz S., Matwin S. (1995)**, “*A WordNet-based Algorithm for Word Sense Disambiguation*”. In In Proceedings of the 14th International Joint Conference on Artificial Intelligence, pp. 1368–1374.
30. **McHale M. (1998)**, “*A Comparison of WordNet and Roget’s Taxonomy for Measuring Semantic Similarity*”. In Proceedings of COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems, pp. 115–120.
31. **Miller G. A. et al. (1990)**, “*Introduction to Wordnet: An On-line Lexical Database*”. International journal of lexicography, vol. 3, no. 4, pp. 235–244.
32. **Mitchell T. M. et al. (2008)**, “*Predicting Human Brain Activity Associated with The Meanings of Nouns*”. Science, vol. 320, no. 5880, pp. 1191–1195.

33. **Nakov P., Popova A., Mateev P. (2001)**, “*Weight Functions Impact on LSA Performance*”. In EuroConference RANLP’2001 (Recent Advances in NLP, pp. 187–193.
34. **Newman D. et al. (2010)**, “*Automatic Evaluation of Topic Coherence*”. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 100–108.
35. **Oflazer K. (1994)**, “*Two-level Description of Turkish Morphology*”. Literary and linguistic computing, vol. 9, no. 2, pp. 137–148.
36. **Oflazer K., Kuruöz I. (1994)**, “*Tagging and Morphological Disambiguation of Turkish Text*”. In Proceedings of the fourth conference on Applied natural language processing. Association for Computational Linguistics, pp. 144–149.
37. **Orhan Z. et al. (2011)**, “*Automated Extraction of Semantic Word Relations in Turkish Lexicon*”. Mathematical and Computational Applications, vol. 16, no. 1, pp. 13.
38. **Patwardhan S., Banerjee S., Pedersen T. (2003)**, “*Using Measures of Semantic Relatedness for Word Sense Disambiguation*”. In Computational linguistics and intelligent text processing. Springer, pp. 241–257.
39. **Patwardhan S., Banerjee S., Pedersen T. (2005)**, “*SenseRelate:: TargetWord: a Generalized Framework for Word Sense Disambiguation*”. In Proceedings of the ACL 2005 on Interactive poster and demonstration sessions. Association for Computational Linguistics, pp. 73–76.
40. **Patwardhan S., Pedersen T. (2006)**, “*Using WordNet-based Context Vectors to Estimate The Semantic Relatedness of Concepts*”. In Proceedings of the EACL 2006 Workshop Making Sense of Sense-Bringing Computational Linguistics and Psycholinguistics Together, vol. 1501, pp. 1–8.
41. **Ponzetto S. P., Strube M. (2007)**, “*Knowledge Derived from Wikipedia for Computing Semantic Relatedness.*” J. Artif. Intell. Res.(JAIR), vol. 30, pp. 181–212.
42. **Poon H., Cherry C., Toutanova K. (2009)**, “*Unsupervised Morphological Segmentation with log-linear Models*”. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of

the Association for Computational Linguistics. Association for Computational Linguistics, pp. 209–217.

43. **Resnik P. (1995)**, “*Using Information Content to Evaluate Semantic Similarity in a Taxonomy*”. In In Proceedings of the 14th International Joint Conference on Artificial Intelligence, pp. 448–453.
44. “<http://tedlab.mit.edu/~dr/SVDLIBC/>”, (Data Download Date: 2011).
45. **Sajib Dasgupta M. K. (2004)**, “*Morphological Parsing of Bangla Words Using Pc-kimmo*”. In Proc. 7 th International Conference on Computer an Information Technology, ICCIT 2004.
46. **Sak H., Güngör T., Saraçlar M. (2007)**, “*Morphological Disambiguation of Turkish Text with Perceptron Algorithm*”. In Computational Linguistics and Intelligent Text Processing. Springer, pp. 107–118.
47. **Sak H., Güngör T., Saraçlar M. (2008)**, “*Turkish Language Resources: Morphological Parser, Morphological Disambiguator and Web Corpus*”. In Advances in natural language processing. Springer, pp. 417–427.
48. **Sak H., Güngör T., Saraçlar M. (2009)**, “*A Stochastic Finite-state Morphological Parser for Turkish*”. In Proceedings of the ACL-IJCNLP 2009 Conference short papers. Association for Computational Linguistics, pp. 273–276.
49. **Sak H., Güngör T., Saraçlar M. (2011)**, “*Resources for Turkish Morphological Processing*”. Language Resources and Evaluation, vol. 45, no. 2, pp. 249–261.
50. **Siblini R., Kosseim L. (2013)**, “*Using a Weighted Semantic Network for Lexical Semantic Relatedness*.” In Angelova G., Bontcheva K., Mitkov R., editors, RANLP. RANLP 2011 Organising Committee / ACL, pp. 610–618.
51. **Strube M., Ponzetto S. P. (2006)**, “*Wikirelate! Computing Semantic Relatedness Using Wikipedia*”. In In Proceedings of the 21st national conference on Artificial intelligence. AAAI Press, pp. 1419–1424.
52. “[http://www2.econ.iastate.edu/classes/crp274/swenson/CRP272/pearsons\\_r\\_and\\_spearman\\_rho.pdf](http://www2.econ.iastate.edu/classes/crp274/swenson/CRP272/pearsons_r_and_spearman_rho.pdf)”, (Data Download Date: 2014).

53. **Takale S. A., Nandgaonkar S. S. (2010)**, “*Measuring Semantic Similarity Between Words Using Web Documents*”. International Journal of Advanced Computer Science and Applications (IJACSA), vol. 1, no. 4, pp. 78–85.
54. **Turdakov D., Velikhov P. (2008)**, “*Semantic Relatedness Metric for Wikipedia Concepts Based on Link Analysis and its Application to Word Sense Disambiguation*”. In Kuznetsov S. D. et al., editors, SYRCoDIS. CEUR-WS, vol. 355 of CEUR Workshop Proceedings.
55. **Turney P. (2001)**, “*Mining The Web for Synonyms: PMI-IR versus LSA on TOEFL*”. In Proceedings of the Twelfth European Conference on Machine Learning, pp. 491–502.
56. “<https://wordnet.princeton.edu/wordnet/man/wnstats.7WN.htmltoc2>”, (Data Download Date: 2014).
57. **Voorhees E. M. (1998)**, “*Using WordNet for Text Retrieval*”. In Fellbaum (Fellbaum, 1998), pp. 285–303.
58. **Wagh K., Kolhe S. (2012)**, “*Information Retrieval Based on Semantic Similarity Using Information Content*”. In IJCSI International Journal of Computer Science Issues, p. 4.
59. “<http://meta.wikimedia.org/wiki/List-of-Wikipedias>”, (Data Download Date: 2014).
60. **Witten I., Milne D. (2008)**, “*An Effective, Low-cost Measure of Semantic Relatedness Obtained from Wikipedia Links*”. In Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA, pp. 25–30.
61. **Wu Z., Palmer M. (1994)**, “*Verbs Semantics and Lexical Selection*”. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, pp. 133–138.
62. **Yeh E. et al. (2009)**, “*WikiWalk: Random Walks on Wikipedia for Semantic Relatedness*”. In Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing. Association for Computational Linguistics, pp. 41–49.
63. **Yuret D., Türe F. (2006)**, “*Learning Morphological Disambiguation Rules for Turkish*”. In Proceedings of the main conference on Human Language



Technology Conference of the North American Chapter of the Association of Computational Linguistics. Association for Computational Linguistics, pp. 328–334.

64. **Zesch T., Gurevych I. (2006)**, “*Automatically Creating Datasets for Measures of Semantic Relatedness*”. In Proceedings of the Workshop on Linguistic Distances. Association for Computational Linguistics, pp. 16–24.
65. **Zesch T., Müller C., Gurevych I. (2008)**, “*Using Wiktionary for Computing Semantic Relatedness.*” In AACL, vol. 8, pp. 861–866.
66. **Zhang W., Feng W., Wang J. (2013)**, “*Integrating Semantic Relatedness and Words’ Intrinsic Features for Keyword Extraction*”. In Proceedings of the Twenty-Third international joint conference on Artificial Intelligence. AAAI Press, pp. 2225–2231.