

DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**DEVELOPING A METHOD FOR AUTOMATIC
CONFIRMATION OF RELATIONSHIPS
BETWEEN CONNOTATION WORDS**



by
Atila SUNCAK

January, 2016
İZMİR

DEVELOPING A METHOD FOR AUTOMATIC CONFIRMATION OF RELATIONSHIPS BETWEEN CONNOTATION WORDS

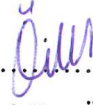
**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
In Computer Engineering Program**

**by
Atilla SUNCAK**

**January, 2016
İZMİR**

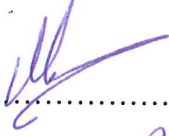
M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “DEVELOPING A METHOD FOR AUTOMATIC CONFIRMATION OF RELATIONSHIPS BETWEEN CONNOTATION WORDS” completed by ATILLA SUNCAK under supervision of ASST. PROF. DR. ÖZLEM AKTAŞ and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



Asst. Prof. Dr. Özlem AKTAŞ

Supervisor



Asst. Prof. Dr. K. Uğur BİRANT

(Jury Member)



Asst. Prof. Dr. Özgün KOŞANER

(Jury Member)



Prof. Dr. Ayşe OKUR

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to thank my supervisor Asst. Prof. Dr. Özlem AKTAŞ for all the supports, patience and encouragement that make me reach success at the end of this influential process.

I would like to thank my friends Sefa KARABAŞ and İnan YILDIRIMLIDAL for their support. I also would like to thank my teacher and my brother Çağdaş Can BİRANT as he believed me more than I did.

This study is developed as a part of the project (Project Number 113E046) supported by TÜBİTAK (The Scientific and Technological Research Council of Turkey), to whom I wish to thank for financial support.

Finally, I would like to offer my special thanks to my family for their support, patience, and help.

Atila SUNCAK

DEVELOPING A METHOD FOR AUTOMATIC CONFIRMATION OF RELATIONSHIPS BETWEEN CONNOTATION WORDS

ABSTRACT

Natural Language Processing (NLP) can be defined as the construction of a computing system that processes and understands natural language. It aims at analyzing the grammar rules of the language and makes it understandable. Frequent use of computer technology resulted in using the software systems that are developed for this purpose such as spelling corrections in text-processing systems.

In this thesis, making computer software determine a semantic relation between two Turkish words is explained. The problem is that computers are incapable of understanding the meaning of a word, unless they are taught. Therefore, using *‘Turkish Synonym and Close-Meaning Dictionary’* and *‘Turkish Antonym Dictionary’* seems to be one of the best solutions to get over this problem.

Definition texts of the words to be compared are split word by word, following that, stop-words which have no specific meaning (articles, pronouns and etc) are omitted. Finally, remaining word groups are compared according to their position as antonym, synonym, close-meaning or similar and the relation is determined by the result of comparison.

As Turkish is an agglutinative language and a word may take affixes in order to produce a noun a verb or an adjective; stemming operation turns out to be the main problem and it makes the situation more complex. Zemberek NLP Library - an open source and platform free software –is applied for morphologic analysis of Turkish words such as stemming, spelling, spell check and etc. Stemming operation of Zemberek assisted to get better results in our comparisons.

By dealing with semantics of words, the study is a contribution to Turkish language research. Besides, one of the objectives of this study is to create the '*Turkish Connotation Dictionary*'.

Keywords: Connotation, antonym, synonym, semantic, stop-word, Turkish



ÇAĞRIŞIM YAPAN KELİMELEER ARASINDAKİ İLİŞKİLERİN ONAYININ OTOMATİKLEŞTİRİLMESİNİ SAĞLAYACAK YÖNTEM GELİŞTİRME

ÖZ

Doğal Dil İşleme, doğal bir dili anlayan ve işleyen yazılım sistemi yapısı olarak tanımlanabilir. Dilin dilbilgisi kurallarını çözümlemeyi ve onu anlaşılabilir kılmayı amaçlar. Bilgisayar teknolojisinin yaygın kullanımı, bu yöntemi kullanan yazılım sistemlerinin geliştirilmesini de beraberinde getirmiştir; metin yazdığımız yazılımlardaki (Microsoft Word gibi) imla hatalarını düzelten mekanizmalar bu yönteme örnektir.

Bu tezde, iki farklı Türkçe sözcük arasındaki anlamsal ilişkinin bilgisayar tarafından saptanması olayı açıklanmıştır. Bilgisayarlar, kendilerine öğretilmediği sürece bir sözcüğün ne anlama geldiğini bilemezler. Bu yüzden, ‘*Türkçe Eş ve Yakın Anlamalı Kelimeler Sözlüğü*’ ile ‘*Zıt Anlamalı Kelimeler Sözlüğü*’ kullanmak, bu sorunu ortadan kaldırmanın en iyi yollarından biri olarak düşünülmüştür.

Karşılaştırılması yapılan kelimelerin tanım metinleri (sözlük karşılıkları) sözcük sözcük ayrıştırılır, daha sonra kendi başlarına anlamları olmayan adıl, ilgeç gibi sözcükler(stop-words) listeden atılır. Son olarak, her bir sözcüğe ait olan, açıklama metinlerinden elde ettiğimiz bu sözcük listelerindeki sözcükler eş, zıt veya yakın anlamlılıklarına göre birbirleriyle karşılaştırılır ve aralarındaki anlamsal ilişkiler belirlenir.

Türkçe, eklemeli bir dil olduğundan, birçok Türkçe sözcük sıfat, ad veya eylem oluşturmak için yapım eki alabilir; bu yüzden kelimeleri kök ve gövdelerine ayırma işlemi, en karmaşık sorun olarak ortaya çıkmakta ve bu durum, olayları daha da karmaşık hale getirmektedir. Açık kaynak kodlu ve platformdan bağımsız bir kütüphane olan Zemberek DDİ(Doğal Dil İşleme) Kütüphanesi, kök-gövde ayırma, heceleme, hata kontrolü gibi biçim bilimsel operasyonları

gerçekleştirebilmektedir. Zemberek karşılaştırmalar sırasında daha iyi sonuçlar alma konusunda oldukça yardımcı olmaktadır.

Bu çalışmanın amaçlarından birisi de ‘*Türkçe Çağrışımlar Sözlüğü*’nü oluşturmaktır. Bu yüzden bu çalışma, aynı zamanda Türkçe dili araştırmalarına da katkı sağlamaktadır.

Anahtar kelimeler: Çağrışım, zıt anlam, eş anlam, semantik, etkisiz kelime, Türkçe



CONTENTS

	Page
M.Sc THESIS EXAMINATION RESULT FORM.....	ii
ACKNOWLEDGEMENTS	iii
ABSTRACT.....	iv
ÖZ	ivi
LIST OF FIGURES	x
LIST OF TABLES	xi
CHAPTER ONE - INTRODUCTION	1
CHAPTER TWO - PREVIOUS WORKS	3
CHAPTER THREE - SYSTEM ARCHITECTURE.....	6
3.1 Used Technologies	6
3.1.1 Visual Studio 2010 Ultimate.....	6
3.1.2 Zemberek NLP Library.....	8
3.1.3 Text Files	9
3.1.3.1 KelimeList.txt	10
3.1.3.2 Açıklama_X.txt	11
3.1.3.3 Results.txt.....	12
3.1.3.4 Stop_Words.txt and usedStopWords.txt	13
3.1.3.5 Bulunamayanlar.txt	14
3.2 Methods of the Algorithm	14
3.2.1 FindEsZit Method	15
3.2.2 GetAciklamaText Method	15
3.2.3 GetAciklamaFile Method	15
3.2.4 GetAciklamaKelimeleriList Method	15
3.2.5 CheckStopWords Method.....	16
3.2.6 GetStemWithZemberek Method.....	16

3.2.7 GetRelations Method	16
CHAPTER FOUR - DEVELOPMENT	18
4.1 Algorithm	18
4.2 Differences in Determination According to Using Stem or Not	24
4.2.1 Using Elements Themselves	24
4.2.2 Using Stem.....	29
CHAPTER FIVE - RESULTS	32
CHAPTER SIX - CONCLUSION	35
REFERENCES.....	37

LIST OF FIGURES

	Page
Figure 2.1 Flow-chart of the ‘Spelling Checker’ algorithm.....	4
Figure 3.1 Microsoft Visual Studio 2010 Ultimate.....	6
Figure 3.2 General view of the project.....	7
Figure 3.3 NZemberek.dll file in the references	8
Figure 3.4 Libraries that ZemberekNLP provides	9
Figure 3.5 Definition Zemberek objects for stemming operations	9
Figure 3.6 Content of KelimeList.txt file.....	11
Figure 3.7 Content of Aciklama_1.txt file	12
Figure 3.8 Content of Results.txt file	13
Figure 3.9 Content of stop_words.txt.....	14
Figure 3.10 Using Zemberek Library in the program	16
Figure 4.1 Flow chart of the algorithm	20
Figure 4.2 Selection-list of a homonym word.....	21
Figure 4.3 Comparing ‘Buzdolabı’ with element-list of ‘Soğuk’	22
Figure 4.4 Comparing ‘Soğuk’ with element-list of ‘Buzdolabı’	23
Figure 4.5 Comparing elements of ‘Soğuk-list’ with elements of ‘Buzdolabı-list’...	23
Figure 4.6 Tedavi and Hastane word couple is compared	25
Figure 4.7 Comparison of Word 1 with the elements of element-list 2.....	27
Figure 4.8 Comparison of Word 2 with the elements of element-list 1	27
Figure 4.9 Comparison of element-list1 with the elements of element-list 2.	28
Figure 4.10 An example of comparison.....	30
Figure 4.11 An example of comparison.....	30

LIST OF TABLES

	Page
Table 2.1 Typical relationships and their examples.....	5
Table 2.2 Relationships and the corresponding patterns.....	5
Table 3.1 Used text files and their purposes	10
Table 4.1 Element list of both words (without stemming or omitting).....	22
Table 4.2 Element list of both words after omitting stop-words (without stemming).....	22
Table 4.3 Element list of both words after omitting stop-words (using stems)	24
Table 4.4 The element-lists of Tedavi and Hastane.....	26
Table 4.5 The element-lists of ‘Using Stems’	29
Table 5.1 The success ratio of ‘using stem’ relation types	32
Table 5.2 The success ratio of ‘using elements themselves’ relation types.....	32
Table 5.3 General view of statistic results of comparisons.....	34

CHAPTER ONE

INTRODUCTION

Due to the rapid development of computer systems in recent years, it is getting harder to pursue the technological breakthrough. It is beyond any doubt that speed of the changes in computer technology amazes everyone. Nowadays, students have their smart boards on classes, doctors can reach any medical history of a patient in no time, and thanks to 3D-Scanners, architects are capable of making their building models easier and without any scale-mistake. In brief, computer technology with the latest developments is being used in every field of life. Once for all, natural languages welcome computer technologies.

Computer technology is already being used in analyzing a language such as translating or spell checking. However, new studies are focused on the lexical and semantic analysis of a language (making computers understand what a word ‘means’ or find the relation between two words according to their meanings).

Developing software on the semantics of a language is a compelling field to study. On the other hand, there are several studies on languages; especially on English. However, it is not possible to benefit an English-oriented algorithm for developing its Turkish version. Natural languages vary among each other due to their grammatical structures such as prefix/suffix rules, conjugation, punctuation and etc. What is more, stop-words which do not have a specific meaning for the reference word such as articles, pronouns, prepositions and etc. in an definition text, are also another obstacle to analyze that language. Therefore, applying an algorithm of a specific language to another one will not function properly.

This study aims at developing an algorithm for Automatic Confirmation of Connotation Words by analyzing their meanings, semantics and comparing them altogether. In order to manage this purpose, some major studies have been used. Main objective is to compare two Turkish words by their meanings, relations among them as antonym/synonym and determine whether they have any meaningful

relationship in common or not. As a result of the determination, the aforementioned words are called as connotation words or not.

Natural Language Processing (NLP), an interaction between natural language and computer as the main purpose is enabling computers to derive semantics and meanings from a natural language, is one of the most challenging subjects of software world in recent years.

Even though countless research has been done, only a handful of successful real life products emerged. Because of the fundamental differences of agglutinative languages i.e. extreme usage of affixes, making NLP research based on those languages is much more difficult. For Turkic Languages situation is even worse (A. A. Akin, & M. D. Akin, 2007, p. 1).

‘Turkish Synonym and Close-Meaning Dictionary’ and ‘Turkish Antonym Dictionary’ are basically word-couple lists that are prepared for Turkish Linguistic Association developed by Dokuz Eylül University, Department of Computer Engineering, Natural Language Processing Work Group (Aktaş et al., 2013). They contain more than 42000 vocabulary couples that have antonym, synonym or close-meaning relations between each other. Following a long-time research, those words have been carefully put together and approved by Turkish Linguistic Association. The dictionaries are used to compare two words to be analyzed and their definition texts.

There are two possible outcomes for comparison of two words according to using stems in the definition texts (word list which contains no stop-word) or not. In most cases, as stems are the smallest parts of a word, using them give the right results. Prefixes and suffixes may change the meaning of the word; therefore it may become hard to find the relations without using stems. On the other hand, in some cases, using the word itself (instead of stem) may also result well. Therefore, study performs both ways and expects a positive outcome from either of them to call these words as ‘connotation words’.

CHAPTER TWO

PREVIOUS WORKS

Turkish is an agglutinative language and frequently uses affixes, and specifically suffixes, or endings (Lewis, 1996). This feature makes things more complex, because almost any word in Turkish gets affixes to produce a different noun, verb, adjective and etc. A suffix can change the word in meaning that is added to and a chain of suffixes can produces several different words in speech, however those final versions of the words are always close to their roots or stems in meaning. Therefore, stemming the word occurs as the main problem. For example:

Çekoslavyahılaştıramadıklarımızdanmışsınız (you are among the one who we could not make from Czechoslovakia)

Çekoslavya-lı-laş-tır-ama-dık-lar-ımız-dan-mış-sınız

Czechoslovakia(stem)-from-become-causative-notable-participle-plural-person1_plural-from-past-person2_plural

The example mentioned above refers that, meanings are changed mostly by suffixes. ‘Zemberek’ library is used to achieve stemming operation.

‘Zemberek NLP Library’ is an open source Turkish NLP library and extension for OpenOffice, LibreOffice (Github, 2015). The library consists plenty of useful operations such as error detection, spelling a word, morphological analyzer, finding the stem and etc.

According to the study performed by Solak and Oflazer (1993), morphological parsing is divided into two as ‘affix stripping’ and ‘root-driven analysis’ methods. The algorithm of the study deals with Turkish grammar rules such as root determination, vowel harmony check, verb/noun parsers and morphophonemic checks. When the mechanism detects any misspelled word or any missing letter of the word, it interprets as ‘incorrect word’. The following flow-chart explains how the algorithm works:

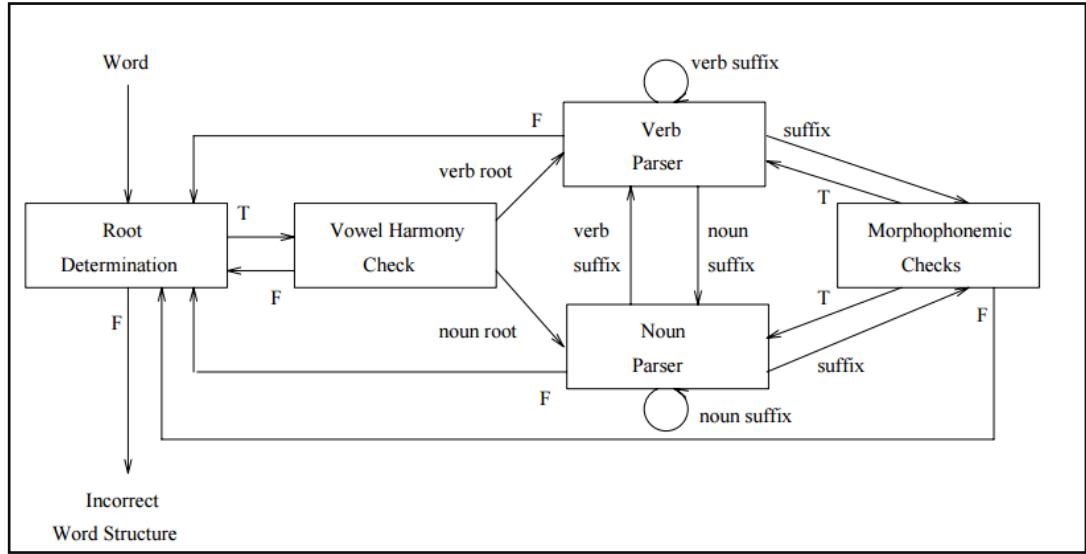


Figure 2.1 Flow-chart of the ‘Spelling Checker’ algorithm

Roots and stems are the smallest meaningful part of the words. The software designed by Birant (2008), introduces a performance for finding roots and suffixes of Turkish words. This study uses a list of possible roots taken from Turkish Linguistic Association and a suffix list taken from The Department of Linguistics. The algorithm follows those steps respectively:

- Start checking from the first letter to the end of the word and compare from the root list whether the character is one of the possible roots or not.
- After finding the possible root, the rest of the word is accepted as a suffix or suffix combination (Birant, 2008).
- Each suffix possibility is controlled from suffix list.
- Checking for the root continues until a valid suffix for the reference root is found.

Another study performed by Orhan, Pehlivan, Uslan and Önder (2011) aims at building an effective lexical-conceptual database according to word relations like synonymy, antonymy, meronymy or hypernymy. The main objective of this study is to understand the meaning of the words and their relations with other ones that is called ‘Rule Extraction’.

Table 2.1 Typical relationships and their examples

RELATION	EXAMPLE
Kind-Of	Fasulye(bean) - bitki(plant)
Amount-Of	Hektar(hectare)-ölçü(measurement)
Group-Of	Manga(squad) –asker(soldier)
Member-Of	Burçak(vetch) –baklagil(leguminous)
Synonym	Ak (White) – Beyaz(White)
Antonymy	Zor (Hard) – Kolay (Easy)

Table 2.2 Relationships and the corresponding patterns

RELATION	RULES	
Kind-Of	Rule1:<X:... Y tipi(dir).> Rule3:<X:... Y türü(dür).>	Rule2:<X:... Y çeşidi(dir).>
Amount-Of	Rule1:<X:... Y birimi(dir).> Rule3:<X:... Y ölçüsü(dür).>	Rule2:<X:... Y miktarı(dır).>
Group-Of	Rule1:<X:... Y topluluğu(dür).> Rule3:<X:... Y birliği(dir).> Rule5:<X:... Y bütünü(dür).> Rule7:<X:... Y sürüsü.>	Rule2:<X:... Y kümesi(dir).> Rule4:<X:... Y (den dan) oluşan topluluk.> Rule6:<X:... Y tümü.>
Member-Of	Rule1:<X:... Y'nin üyesi(dir).> Rule3:<X:... Y sınıfı.>	Rule2:<X:... Y +gillerden(dir).> Rule4:<X:... Y takımı.>
Synonym	Rule1:<X: Y (single word).>	Rule2:<X:... Y. (after comma, the last word>
Antonym	Rule1:<X:... Y karşıtı.>	Rule2:<X:... Y olmayan.>

CHAPTER THREE

SYSTEM ARCHITECTURE

This project is a Windows Form Application via Visual Studio 2010 Ultimate and written by C# programming language. Several text files are used for specific purposes and Zemberek NLP library is used for stemming words.

3.1 Used Technologies

3.1.1 Visual Studio 2010 Ultimate

Visual Studio is an IDE (Integrated Development Environment) designed by Microsoft. It enables developing websites, computer programs, form applications, web services and etc. In the performance of this thesis, Windows Form Application is used as project platform.

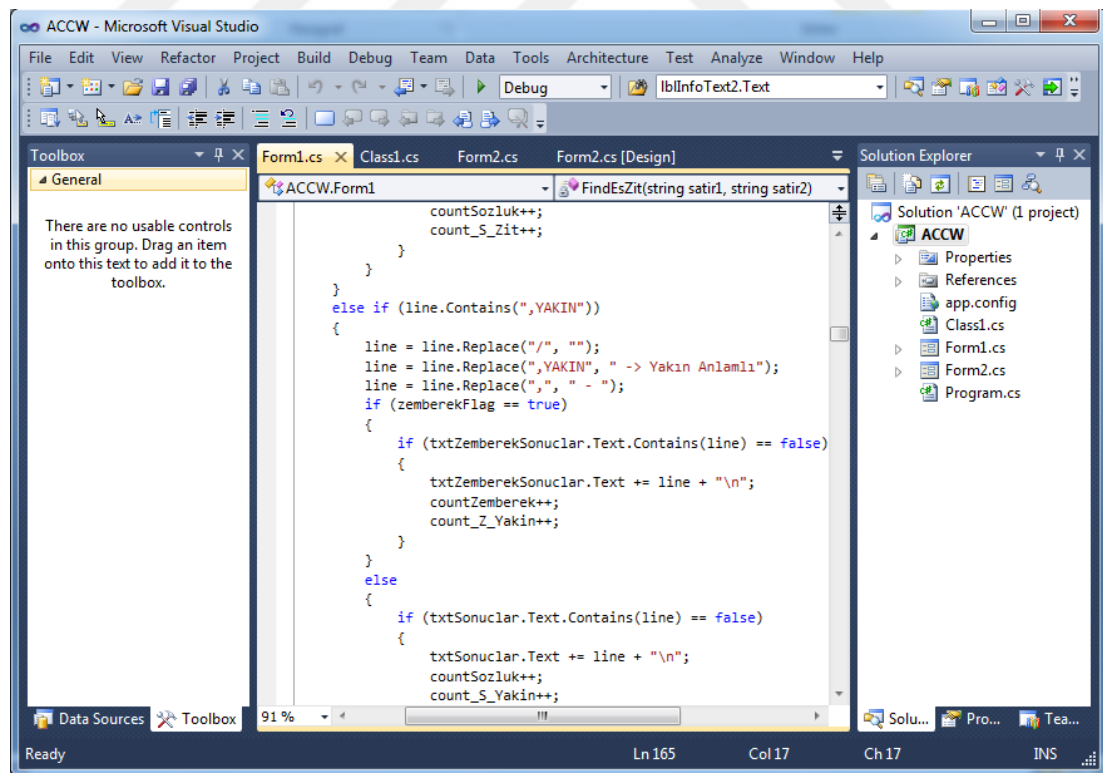


Figure3.1 Microsoft Visual Studio 2010 Ultimate

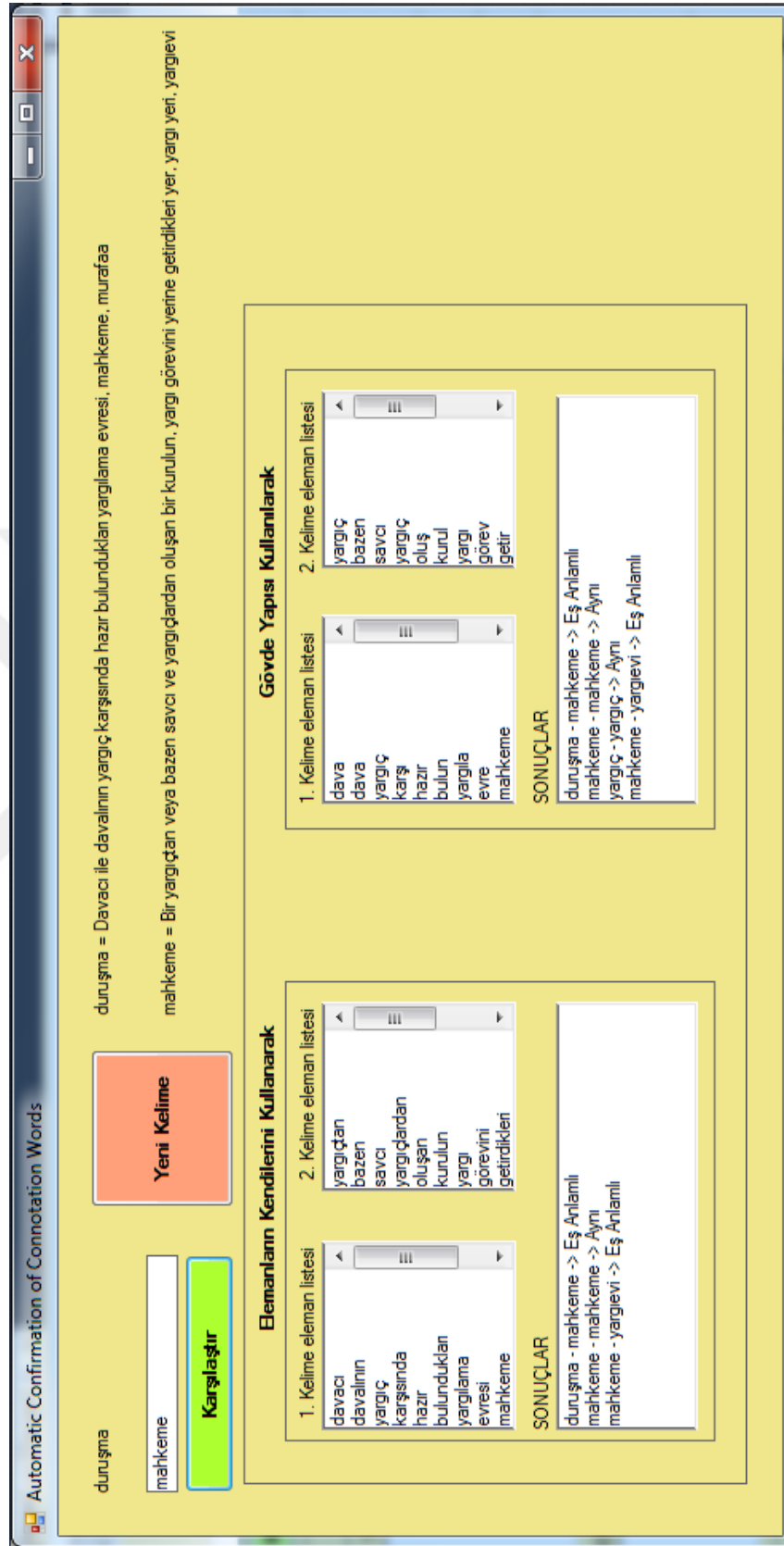


Figure 3.2 General view of the project

3.1.2 Zemberek NLP Library

Zemberek NLP Library - an open source and platform free software - is used for morphologic analysis of Turkish words such as stemming, spelling, spell check and etc.

Zemberek is supported by Visual Studio to be used in the project. The following processes explain what operations have to be handled before using Zemberek in the software:

- The library folder is downloaded (Nzemberek, 2015)
- NZemberek.dll file is added to the IDE as reference shown in Figure 3.3.
- The relevant 'using' identifiers are added to the project as shown in Figure 3.4.
- The necessary objects are defined as shown in Figure 3.5.

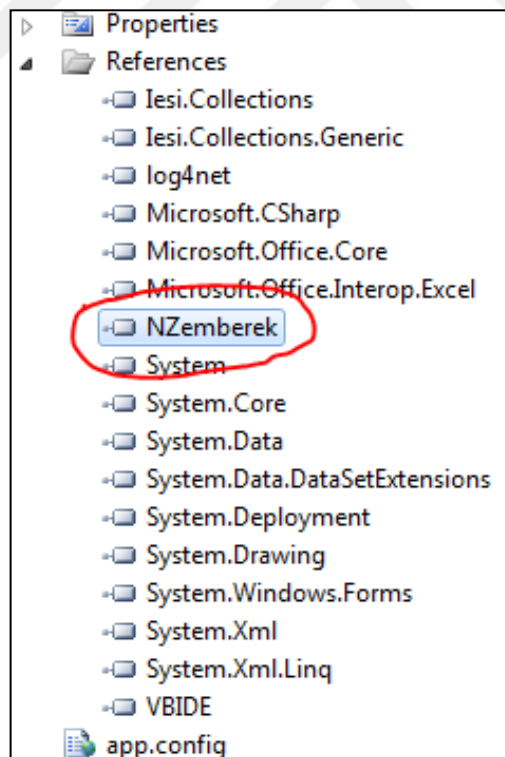


Figure 3.3 NZemberek.dll file in the references

```

using net.zemberek.erisim;
using log4net;
using net.zemberek.yapi;
using net.zemberek.yapi.ek;
using net.zemberek.bilgi.kokler;
using net.zemberek.yapi.kok;
using net.zemberek.tr.yapi;
using net.zemberek.tr.yapi.ek;
using net.zemberek.tr.yapi.kok;
using net.zemberek.islemler.cozumleme;
using net.zemberek.islemler;
using net.zemberek.tr.islemler;
using net.zemberek.bilgi.araclari;
using net.zemberek.bilgi;

```

Figure 3.4 Libraries that ZemberekNLP provides

```

public string GetStemWithZemberek(string giris)
{
    string kelimeson = "";
    Zemberek zemberek = new Zemberek(new TurkiyeTurkcesi());
    Kelime[] cozumler = zemberek.kelimeCozumle(giris);
    Kelime kelime1 = new Kelime();
    try
    {
        kelime1 = cozumler[0];
        kelimeson = kelime1.kok().icerik();
    }
    catch (Exception ex)
    {
        ex.Message.ToString();
        kelimeson = giris;
    }

    return (kelimeson);
}

```

Figure 3.5 Definition Zemberek objects for stemming operations

3.1.3 Text Files

There are several files -which are the data storages of this software - used for specific purposes such as saving results, checking for word couples, getting lexical meanings of the words and etc. All the files are listed and explained respectively as shown in Table 3.1.

Table 3.1 Used text files and their purposes

FILE NAME	EXPLANATION
KelimeList.txt	Includes antonym, synonym and close-meaning word couples
Aciklama_1.txt	Includes definition-texts of the words whose first character is a/b/c/ç/d
Aciklama_2.txt	Includes definition-texts of the words whose first character is e/f/g/h/ı/i/j
Aciklama_3.txt	Includes definition-texts of the words whose first character is k/l/m/n/o/ö/p
Aciklama_4.txt	Includes definition-texts of the words whose first character is r/s/ş/t/u/ü/v/y/z
Results.txt	Includes the statistical result data
stop_words.txt	The list of most frequently used stop-words of Turkish
usedStopWords.txt	Includes the stop-words that have been come across during performance.
bulunamayanlar.txt	Includes the word couple list which the system could not find any relationship between each other

3.1.3.1 KelimeList.txt

‘KelimeList.txt’ file is a combination of word-couple lists with more than 42000 elements retrieved from ‘Turkish Synonym and Close-Meaning Dictionary’ and ‘Turkish Antonym Dictionary’ prepared for Turkish Linguistic Association developed by Dokuz Eylül University, Department of Computer Engineering, Group of Natural Language Processing (Aktaş et al., 2013).

Every word has both its relevant one and the type of relation such as synonym, antonym or close-meaning. Like other languages, Turkish also has homonyms like ‘kara’ which has several different meanings such as ‘siyah renkli (*black*)’ or ‘toprak parçası (*earth*)’. This issue has also been considered during the process of preparation of the list:

- Kara (black) – beyaz (white) – Zıt anlam (Antonym)
- Kara (earth) – toprak(earth, soil) – Eşanlam (Synonym)

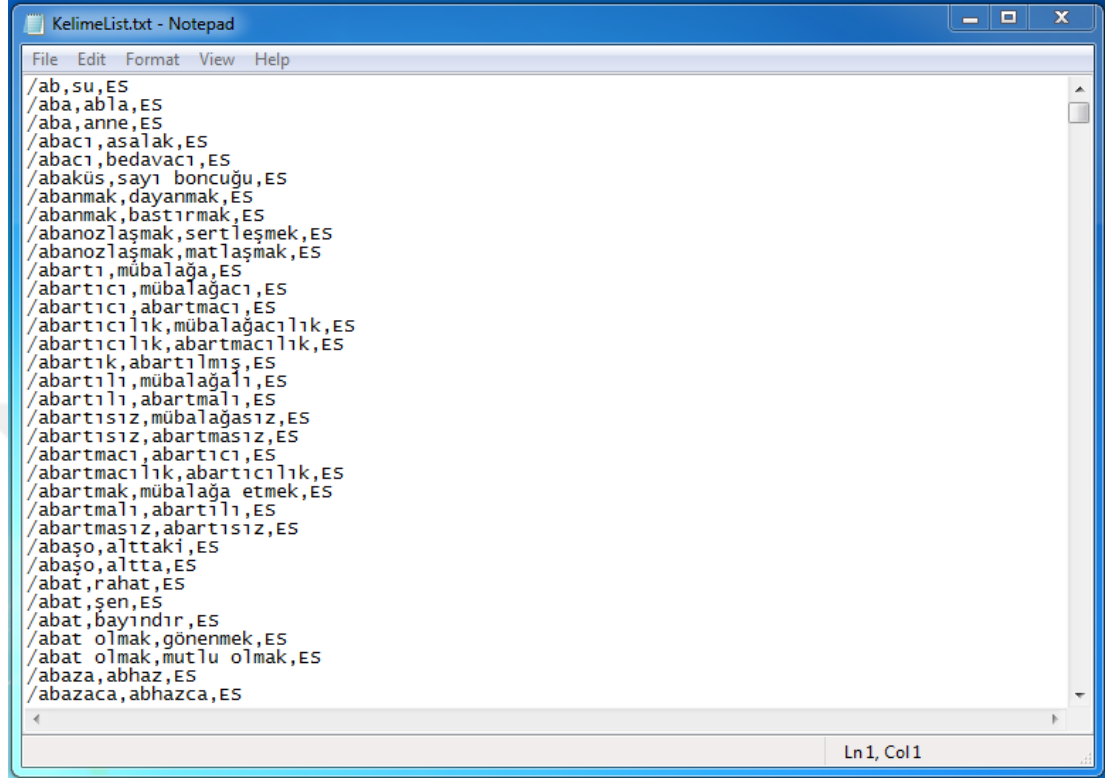


Figure 3.6 Content of kelimeList.txt file

3.1.3.2 Açıklama_X.txt

‘Açıklama_X.txt’ file is a list which contains more than 115000 Turkish words with their lexical meanings. However, in order to increase the performance of the software, the list file is divided into 4 separate sub-files according to initials of the words:

- Açıklama_1.txt includes the words with the initials a,b,c,ç and d.
- Açıklama_2.txt includes the words with the initials e,f,g,h,ı,i and j.
- Açıklama_3.txt includes the words with the initials k,l,m,n,o,ö and p.
- Açıklama_4.txt includes the words with the initials r,s,ş,t,u,ü,v,y and z.

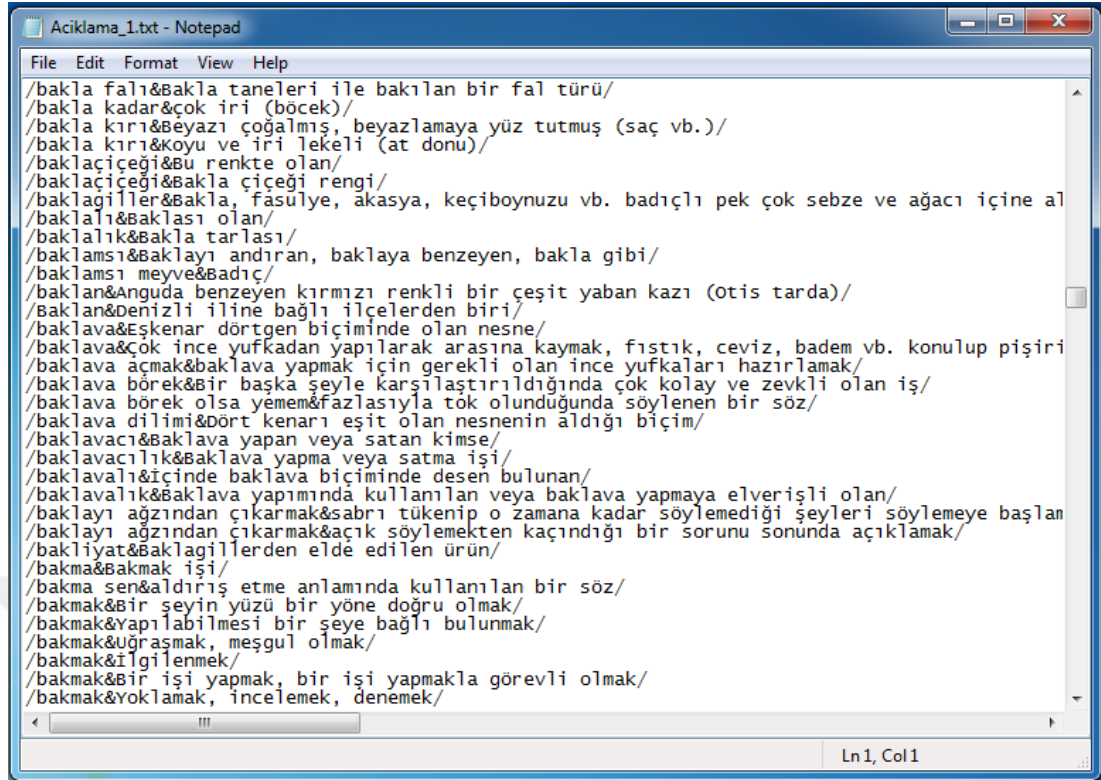


Figure 3.7 Content of aciklama_1.txt file

3.1.3.3 Results.txt

Following every process in the software, the statistical consequences are held in 'Results.txt' file. In comparison process, using words themselves or their stems may result differently. In every comparison, the number of the following relations are respectively calculated and held in two groups as the relationship types also differ according to the comparing type:

- Total relationships using stems
 - Total synonym relationships
 - Total antonym relationships
 - Total close-meaning relationships
 - Total same-words
- Total relationships without using stems
 - Total synonym relationships

- Total antonym relationships
- Total close-meaning relationships
- Total same-words

File	Edit	Format	View	Help								
mikyas	ölçek	2	4	1	0	0	1	2	0	0	2	
kederlenmek	dertlenmek	2	2	2	0	0	0	0	0	2	0	
öbürleri	diğerleri	2	1	1	0	0	0	0	1	1	0	
kıyı	kenar	1	1	1	0	0	0	1	0	0	0	
mir	alay	0	0	0	0	0	0	0	0	0	0	
çokları	bazıları	0	0	0	0	0	0	0	0	0	0	
haytalık	serseri	3	6	2	0	0	1	3	0	0	0	
eksilmek	azalmak	3	2	1	0	0	2	1	0	0	0	
sahileştirmek	gerçekleştirmek	4	3	2	0	0	0	2	1	0	0	
karı	koca	5	9	3	0	0	2	5	0	0	4	
komutan	asker	1	2	0	0	0	1	1	0	0	1	
kaz	ördek	0	0	0	0	0	0	0	0	0	0	
sanıvermek	zannetmek	3	1	2	0	0	0	1	0	0	0	
profesyonel	acemi	2	3	1	0	0	1	1	0	0	0	
ajan	casus	4	2	0	1	1	1	0	0	0	1	
arlanmaz	utanmaz	8	3	6	0	0	0	2	1	0	0	
fazlalaşmak	azalmak	0	0	0	0	0	0	0	0	0	0	
bilgilendirmek	bilgi vermek	0	0	0	0	0	0	0	0	0	0	
çirkin	güzel	2	2	1	0	0	1	1	0	0	0	
arz	talep	0	0	0	0	0	0	0	0	0	0	
nazizm	nazi	0	0	0	0	0	0	0	0	0	0	
nazizm	nazi	0	0	0	0	0	0	0	0	0	0	
gerek	mecbur	0	0	0	0	0	0	0	0	0	0	
itinalı	özenli	3	3	1	0	0	2	0	0	0	1	
atmak	tutmak	0	0	0	0	0	0	0	0	0	0	
acele etmek	çabuk	2	2	1	0	0	1	1	0	0	0	
lokum	tatlı	1	1	1	0	0	0	0	0	0	1	
geri	ileri	2	2	0	1	0	1	0	1	0	1	
soy	kütük	0	0	0	0	0	0	0	0	0	0	
kirlilemek	temizlemek	0	0	0	0	0	0	0	0	0	0	
bacak	ayak	1	1	0	0	0	1	0	0	0	0	
şarkı	müzik	0	1	0	0	0	0	0	0	0	1	
nesnel	objektif	4	4	2	0	0	2	2	0	0	0	
geniş	dar	1	1	0	1	0	0	1	0	0	0	

Figure 3.8 Content of results.txt file

3.1.3.4 Stop_Words.txt and usedStopWords.txt

During the process of comparison, definition texts of the compared words are analyzed. In the favor of the best result, the stop-words have to be omitted as they have no specific meaning for the referenced word and may mislead the final evaluations of comparison. Therefore, the basic stop-words of Turkish language are collected in the *stopWords.txt* file for this objective.

UsedStopWords.txt file collects the stop-words found in the definition texts during comparison. Those held data help us figure out the statistics of the commonly used stop-words.

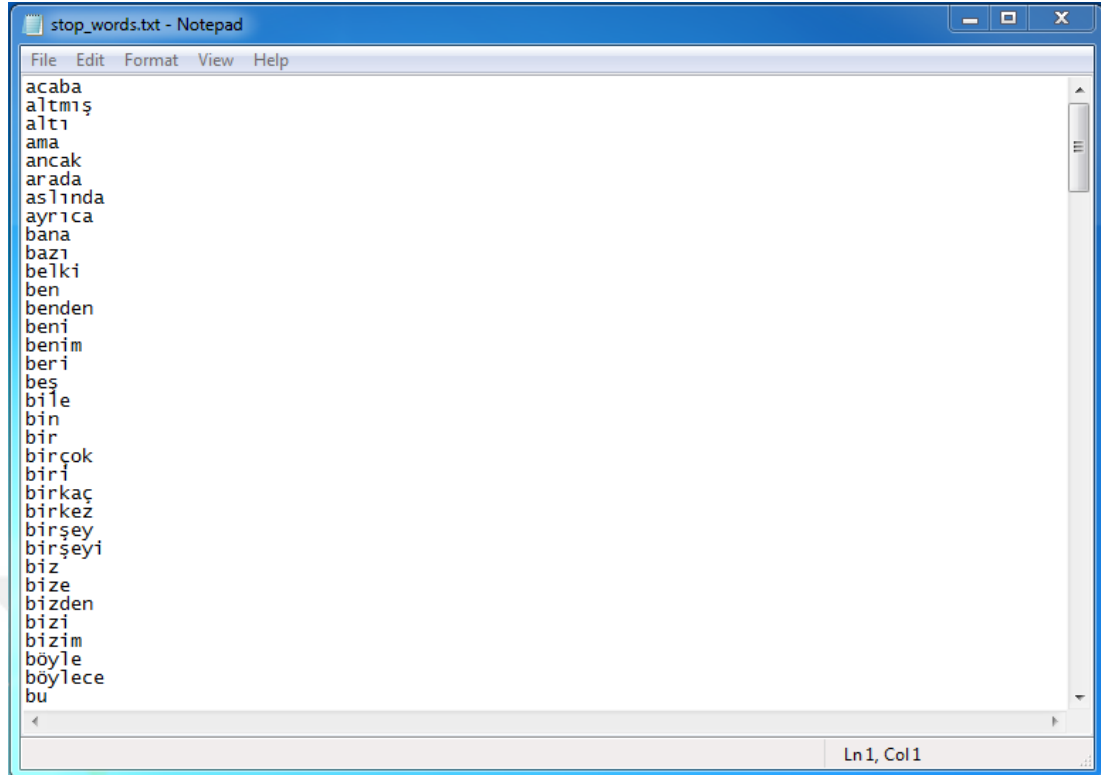


Figure 3.9 Content of stop_words.txt

3.1.3.5 *Bulunamayanlar.txt*

‘Bulunamayanlar.txt’ file is used for saving the word couples which the system could not determine any relationship between each other. During comparison, when no relationship is found, at this point, the system checks this list in case of encountering the same word couple again. If a word couple is not determined any relationship between each other and the system find the same word couple in this file, then there may be a possibility of an undetermined relationship so it needs to be analyzed manually.

3.2 Methods of the Algorithm

In the project, there are some important methods for performing specific purposes. These methods are briefly explained below:

3.2.1 FindSynAcr Method

‘FindSynAcr’ method is used for searching a relationship for the word couple in the KelimeList.txt file.

3.2.2 GetInfoText Method

‘GetInfoText’ method is used for getting the definition texts of the words from appropriate Aciklama_X.txt file.

3.2.3 GetInfoFile Method

‘GetInfoFile’ method determines which Aciklama_X.txt file should be used according to the initial of the word.

3.2.4 GetElementList Method

‘GetElementList’ method is used for creating the element lists of both words to be compared. Each word’s definition text is separated word by word (elements). In the performance of stemming method, at first each element’s stems are found, then those stems are listed. However, in the performance of non-stemming method, those elements remain the same and listed as their original versions. As a result, 4 different lists are obtained:

- Word1 -> list of stems of definition text elements
- Word2 -> list of stems of definition text elements
- Word1 -> list of elements of definition text
- Word2 -> list of elements of definition text

3.2.5 CheckStopWords Method

‘CheckStopWords’ method is used for checking the stop-words of both definition texts. When a stop-word is found (by comparing with list instop_words.txt), the stop-word is omitted from the element list. Furthermore, the stop-word is stored in usedStopWords.txt file for statistical operations.

3.2.6 GetStemWithZemberek Method

‘GetStemWithZemberek’ method returns the stem of the given word by using Zemberek NLP Library.

```
public string GetStemWithZemberek(string giris)
{
    string kelimeson = "";
    Zemberek zemberek = new Zemberek(new TurkiyeTurkcesi());
    Kelime[] cozumler = zemberek.kelimeCozumle(giris);
    Kelime kelime1 = new Kelime();
    try
    {
        kelime1 = cozumler[0];
        kelimeson = kelime1.kok().icerik();
    }
    catch (Exception ex)
    {
        ex.Message.ToString();
        kelimeson = giris;
    }

    return (kelimeson);
}
```

Figure 3.10 Using Zemberek Library in the program

3.2.7 GetRelations Method

‘GetRelations’ method compares all the listed and separated words and elements with each other in a specific order. As similar word/element in both lists also shows relationship between the words to be compared, the method also checks whether any similar elements are detected or not. The comparison steps are listed below:

- Word1 with Word2's elements
- Word1's elements with Word2
- Word1's elements with Word2's elements
- If there is still no match, then Word1-Word2 couple is searched in bulunamayanlar.txt file.



CHAPTER FOUR

DEVELOPMENT

Due to of the fact that semantic analysis requires some expert system methods, natural language processing is one of the most difficult areas of software technology. A decision mechanism and a simple machine learning method are performed in the study. In addition, in order to get reliable results after whole comparisons, stemming operation is needed.

‘Turkish Antonym Dictionary’ and ‘Turkish Synonym and Close-Meaning Dictionary’ are used for comparison of words. A list that includes more than 42000 word couples and another list of words with their lexical meanings which includes more than 115000 words are used.

Turkish language has many homonyms which are written the same but have different meanings from each other such as the word ‘kara’ that means both ‘earth’ and ‘black’. In order to prevent confusion in the meanings before comparison; the user is required to select the desired meaning from the selection-list when the proposed word has a homonym.

4.1 Algorithm

The steps of the algorithm are listed below:

- The word which the user writes is checked whether it has a homonym or not and in case of homonymy, the user is required to select the right meaning from the selection-list.
- Word1 and word2 are compared by using the list of word couples that have semantic relationships such as antonym, synonym or close-meaning.
- Following the comparison, definition texts of each word are found from definition-text list.
- Each element (words) of definition-texts is separated word by word.

- Results are compared according to using stems or the elements themselves, four separate lists are created by software:
 - Word1's elements themselves
 - Word2's elements themselves
 - Stems of word1's elements
 - Stems of word2's elements
- Stemming operations are performed by using Zemberek NLP Library.
- All stop-words in the lists are found and omitted.
- Word1 is compared with each word2's elements (as in Step1)
- Word2 is compared with each word1's elements (as in Step1)
- Each word1's elements are compared with each word2's elements (as in Step1)
- When any match could not be found, word1-word2 couple is checked in bulunamayanlar.txt file (all unrelated word couples are stored in that file). By performing this mechanism, a previously unrelated word couple may be found again.
- When all comparisons become unsuccessful, that word couple is stored in 'bulunamayanlar.txt' file.

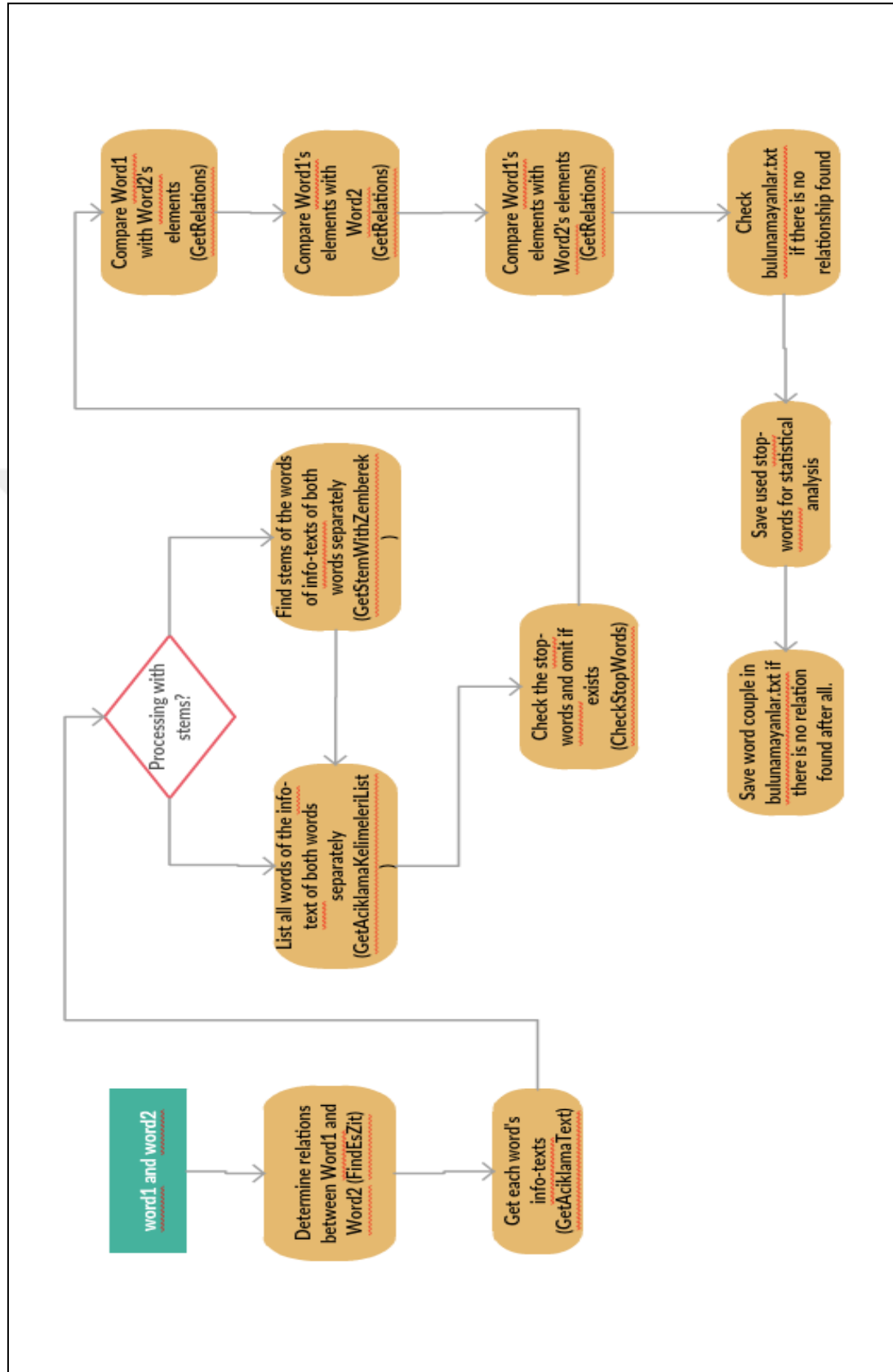


Figure 4.1 Flow chart of the algorithm

In Figure 4.2, the selection-list panel for a homonym is shown. The word ‘göz(eye)’ proposed as Word2 by the user, however the right meaning of it has to be chosen as ‘göz’ is a homonym word. The selection-list includes all meanings for the aforementioned homonym.

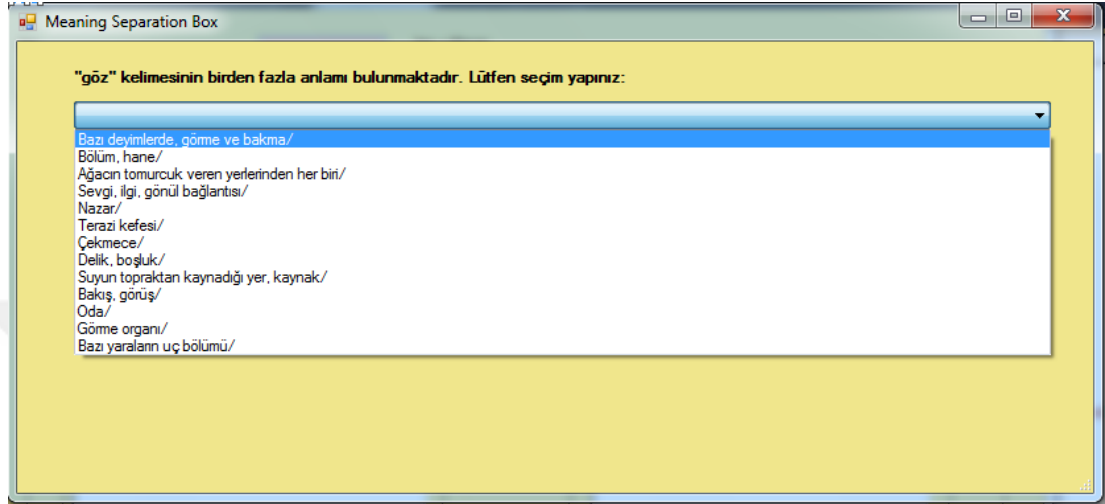


Figure 4.2 Selection-list of a homonym word

In order to motivate the set of operation to determine the relationship between two words, ‘Buzdolabı’ (the first word) and ‘Soğuk’ (the second word) is examined below:

- Buzdolabı : Yiyecek ve içecekleri soğuk tutmaya yarayan, motorla çalışan dolap
- Soğuk: Duygudan ve sevgiden yoksun olan, yakın ve içten olmayan, ilgisiz

In the first step of examination, ‘buzdolabı’ and ‘soğuk’ are compared from ‘Turkish Antonym Dictionary’ and ‘Turkish Synonym and Close-Meaning Dictionary’. As a result of checking, there is not any antonym, synonym or close-meaning relation found. Following that, element-lists are created as seen in Table4.1.

Table 4.1 Element list of both words (without stemming or omitting)

BUZDOLABI	SOĞUK
yiyecek	duygudan
Ve	ve
içecekleri	sevgiden
soğuk	yoksun
tutmaya	olan
yarayan	yakın
motorla	ve
çalışan	içten
dolap	olmayan
	ilgisiz

In element-list1 ‘ve’ and yarayan; in element-list2 ‘ve’, ‘olan’ and ‘olmayan’ are stop-words. Therefore they are omitted from both lists as it is seen in Table 4.2.

Table 4.2 Element list of both words after omitting stop-words (without stemming)

BUZDOLABI	SOĞUK
yiyecek	duygudan
içecekleri	sevgiden
soğuk	yoksun
tutmaya	yakın
motorla	içten
çalışan	ilgisiz
dolap	

‘Buzdolabı’ with words of element-list2, ‘soğuk’ with words of element-list1 and elements of buzdolabı-list with elements of soğuk-list are located below as it is seen in Figure 4.3, Figure 4.4 and Figure 4.5

<p>BUZDOLABI - duygudan</p> <p>BUZDOLABI - sevgiden</p> <p>BUZDOLABI - yoksun</p> <p>BUZDOLABI - yakın</p> <p>BUZDOLABI - içten</p> <p>BUZDOLABI - ilgisiz</p>
--

Figure 4.3 Comparing ‘Buzdolabı’ with element-list of ‘Soğuk’

SOĞUK - yiyecek
SOĞUK - içecekleri
SOĞUK - soğuk
SOĞUK - tutmaya
SOĞUK - motorla
SOĞUK - çalışan
SOĞUK - dolap

Figure 4.4 Comparing ‘Soğuk’ with element-list of ‘Buzdolabı’

duygudan – yiyecek duygudan – içecekleri duygudan – soğuk duygudan – tutmaya duygudan – motorla duygudan – çalışan duygudan – dolap sevgiden – yiyecek sevgiden – içecekleri sevgiden – soğuk sevgiden – tutmaya sevgiden – motorla sevgiden – çalışan sevgiden – dolap yoksun – yiyecek yoksun – içecekleri yoksun – soğuk yoksun – tutmaya yoksun – motorla yoksun – çalışan yoksun – dolap	yakın – yiyecek yakın – içecekleri yakın – soğuk yakın – tutmaya yakın – motorla yakın – çalışan yakın – dolap içten – yiyecek içten – içecekleri içten – soğuk içten – tutmaya içten – motorla içten – çalışan içten – dolap ilgisiz – yiyecek ilgisiz – içecekleri ilgisiz – soğuk ilgisiz – tutmaya ilgisiz – motorla ilgisiz – çalışan ilgisiz – dolap
---	--

Figure 4.5 Comparing elements of ‘Soğuk-list’ with elements of ‘Buzdolabı-list’

Following the comparison operations, the software is expected to find those two relations:

- Soğuk (word2) – soğuk(element-list1) => same words
- Soğuk(element-list1) – ilgisiz(element-list2) => synonym

Aforementioned relationships are determined as a result of comparison processes without using stems. At this stage, the performance of comparisons is held using stems.

Table 4.3 Element list of both words after omitting stop-words (using stems)

BUZDOLABI	SOĞUK
yiyecek	duygu
içecek	sevgi
soğuk	yoksun
tut	yakın
motor	içten
çalış	ilgi
dolap	

Using stems clearly changes the words in both element-lists and it certainly affects the comparison and the result. The set of comparison operations are performed respectively by using ‘Turkish Antonym Dictionary’ and ‘Turkish Synonym and Close-Meaning Dictionary’. Finally the software finds only one relation:

- Soğuk (word2) – soğuk(element-list1) => same words

As the word ‘ilgisiz’ is replaced by its stem ‘ilgi’, the word ‘soğuk’ is compared with it and naturally there is no relation between them.

Example shows that using stem may affect the result. Nevertheless, there have been relationships found in both cases which show that those two words (soğuk, buzdolabı) are connotation words.

4.2 Differences in Determination According to Using Stem or Not

4.2.1 Using Elements Themselves

In Figure 4.6, the word ‘tedavi (treatment)’ is randomly selected by software, and the word ‘hastane (hospital)’ is proposed by the user as a connotation word. While the random word comes into the screen, its definition text is also shown for the user to make the inference easier. Following clicking ‘Karşılaştır’ button, the element-list is created and the results for both cases are shown on the screen.

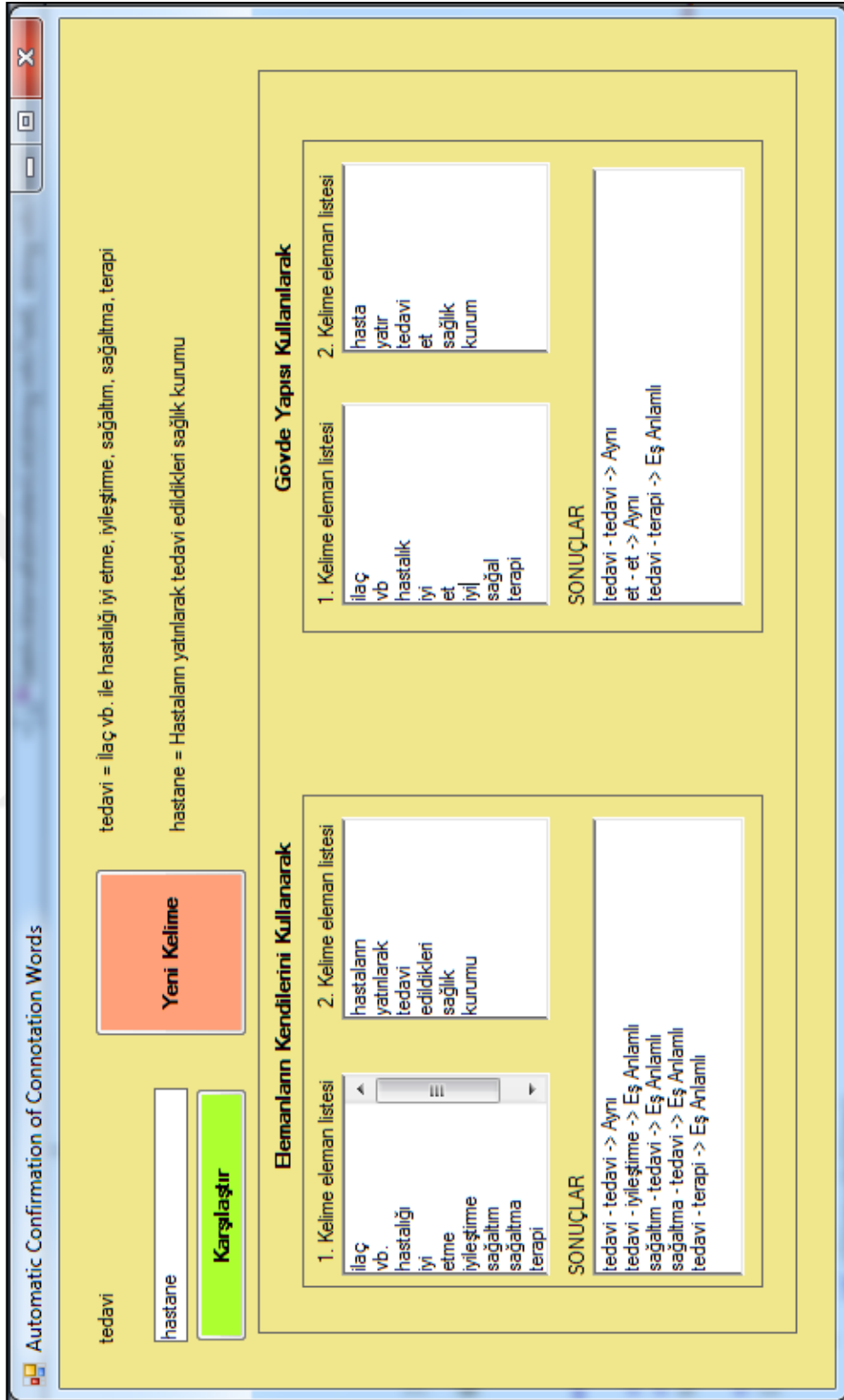


Figure 4.6 Tedavi and Hastane word couple is compared

Because software performs the determination for both cases, for each case two couples of element-lists are determined and listed.

- Tedavi: İlaç vb. ile hastalığı iyi etme, iyileştirme, sağaltım, sağaltma, terapi.
- Hastane: Hastaların yatırılarak tedavi edildikleri sağlık kuruluşu

Table 4.4 The element-lists of Tedavi and Hastane

TEDAVİ	HASTANE
ilaç	hastaların
vb.	yatırılarak
hastalığı	tedavi
iyi	edildikleri
etme	sağlık
iyileştirme	kurumu
sağaltım	
sağaltma	
terapi	

As it is seen in Table 4.4, the words (elements) of both definition texts are listed with no change. On the other hand, there is one word missing in the first list. ‘İle (with)’ is not located in the list because it is a stop-word which does not have specific meaning for TEDAVİ (TREATMENT). As no stop-word in the second definition text is found, there is no missing word in the second element-list.

The next part of the determination is the comparison operations. As it is mentioned before, there is an order of performance:

- Word 1 with the elements of element-list 2.
- Word 2 with the elements of element-list 1.
- The elements of element-list1 with the elements of element-list 2.
- Check ‘bulunamayanlar.txt’ file.

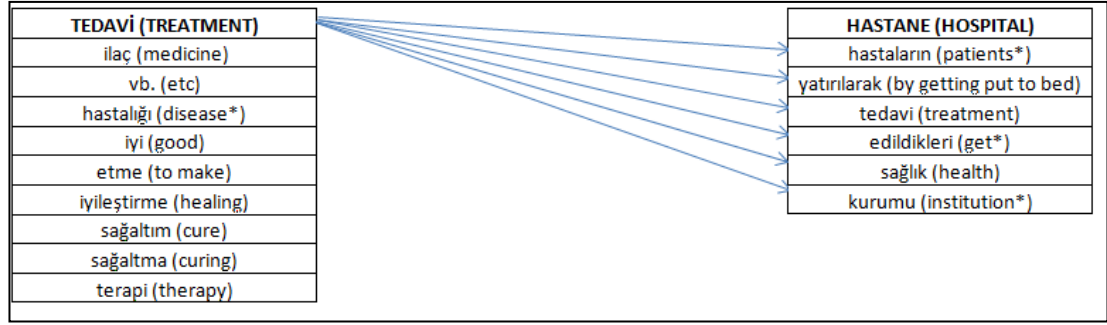


Figure 4.7 Comparison of Word 1 with the elements of element-list 2

As shown in Figure 4.7, one relation is found between Word1 (TEDAVİ) and element-list2:

- Tedavi – tedavi (same word)

Relation is held in a list (result-list) and the other comparisons go on performing. In every comparison, relations found are stored in the result-list but while performing this operation, as there is no point in storing the same relations, the software also checks whether the list includes that relation or not. There are four relation types which are Synonym, Antonym, Close-Meaning and Same Words.

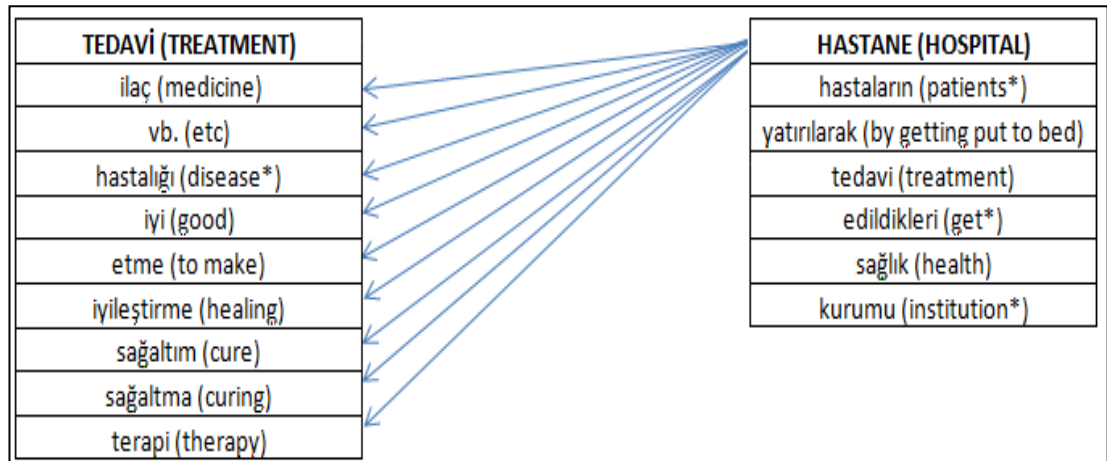


Figure 4.8 Comparison of Word 2 with the elements of element-list 1

When comparing word 2 (HASTANE) and element-list1 as seen in Figure 4.8, software finds no relation.

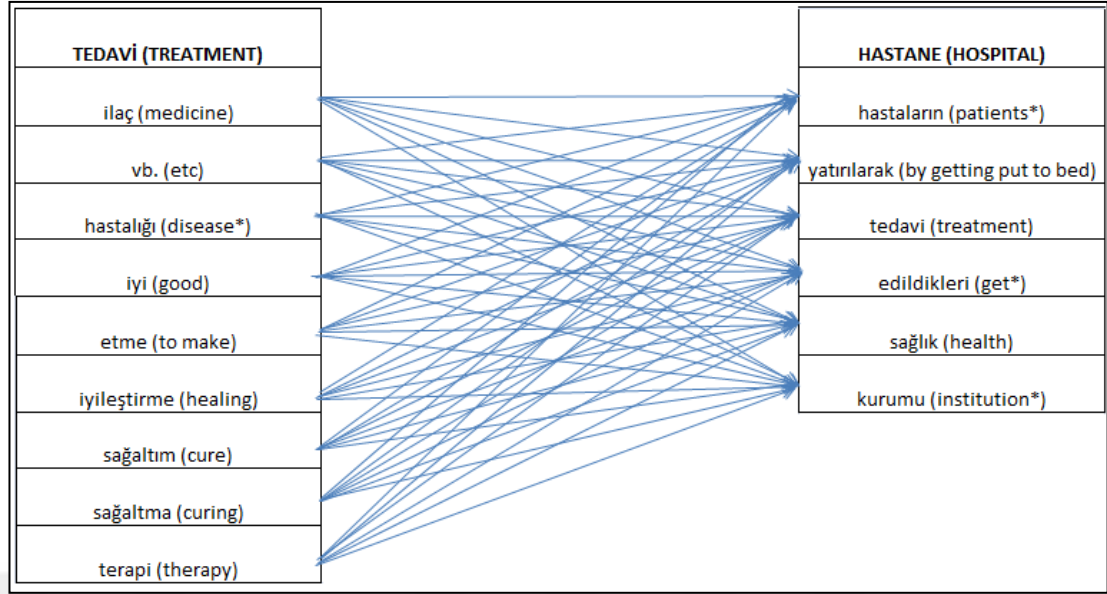


Figure 4.9 Comparison of element-list1 with the elements of element-list 2.

Comparing elements of both element-lists with each other produces generally more relations than other comparison parts. As Figure 4.9 shows, there are a lot of comparisons which varies according to the number of elements in an element-list. In example, the software found four relations:

- tedavi – iyileştirme (synonym)
- sağaltım – tedavi (synonym)
- sağaltma – tedavi (synonym)
- tedavi – terapi (synonym)

Finally, all the comparisons are over and all relations are listed in the result-list:

- Tedavi – tedavi (same)
- tedavi – iyileştirme (synonym)
- sağaltım – tedavi (synonym)
- sağaltma – tedavi (synonym)
- tedavi – terapi (synonym)

4.2.2 Using Stem

In addition to the previous case, software also performs all of the operations by using stems of the words. In Table 4.5, the element-lists are represented according to stems:

Table 4.5 The element-lists of ‘Using Stems’

TEDAVİ	HASTANE
ilaç	hasta
vb	yatır
hastalık	tedavi
iyi	et
et	sağlık
sağal	kurum
terapi	

It is clearly seen that there are some major changes in the list. In the first element-list, two lines are removed: ‘iyileştirmek’ is generated from the stem ‘iyi’; as there is already a word ‘iyi’ in the list, the second one is removed; ‘sağaltmak’ and ‘sağaltım’ are both generated from the stem ‘sağal’, therefore one of them is also removed. Furthermore, some words are replaced by their stems such as ‘hastaların’ with ‘hasta’. All stem-check operations are performed by using ZemberekNLP Library.

Following those changes, the set of comparison operations are performed and the found relations are stored in a second result-list:

- tedavi – tedavi (same)
- et – et (aynı)
- tedavi – terapi (synonym)

In the example, as there are relationships in both cases, software interprets as those two words are connotation words. Although using stems resulted fewer relationships than the other case, it does not mean that using elements themselves is the best option. Statistical results, scrutinized in next chapter, shows that using stems

result more relationships. In some examples, there may be no relationship in using-stem case or vice versa. It is all about the contents of the definition texts as the element-lists are consists of the elements from the definition texts.

The screenshot shows the 'Automatic Confirmation of Connotation Words' window. At the top, the word 'millet' is entered, and its definition is 'millet = Bir yerde bulunan kimselerin bütünü, herkes'. Below it, 'ülke' is entered, and its definition is 'ülke = Bir devletin egemenliği altında bulunan toprakların tümü, diyar, memleket'. A 'Yeni Kelime' button is visible. The 'Karşılaştır' button is highlighted. Below the input fields, there are two main sections: 'Elemanların Kendilerini Kullanarak' and 'Gövde Yapısı Kullanılarak'. Each section contains two lists of elements and a 'SONUÇLAR' (Results) box. In the 'Elemanların Kendilerini Kullanarak' section, the results are empty. In the 'Gövde Yapısı Kullanılarak' section, the results show 'yer - ülke -> Eş Anımlı' and 'tüm - bütün -> Eş Anımlı'.

Figure 4.10 An example of comparison

‘Millet’ and ‘ülke’ words are compared in Figure 4.10. As it is seen, while using elements themselves resulted in no relationship, using stems brought two relations.

The screenshot shows the 'Automatic Confirmation of Connotation Words' window. At the top, the word 'fan' is entered, and its definition is 'fan = Havalandıma aracı, pervane, pervane kanadı, vantilatör'. Below it, 'havalandıma' is entered, and its definition is 'havalandıma = Kapalı bir yerin havasını değiştirmek amacıyla dışardan temiz hava girişini veya çeşitli'. A 'Yeni Kelime' button is visible. The 'Karşılaştır' button is highlighted. Below the input fields, there are two main sections: 'Elemanların Kendilerini Kullanarak' and 'Gövde Yapısı Kullanılarak'. Each section contains two lists of elements and a 'SONUÇLAR' (Results) box. In the 'Elemanların Kendilerini Kullanarak' section, the results show 'havalandıma - havalandıma -> Aynı'. In the 'Gövde Yapısı Kullanılarak' section, the results are empty.

Figure 4.11 An example of comparison

Figure 4.11 shows the comparison of ‘fan’ and ‘havalandırma’ words. In the case of not using stems found the aforesaid relationship. The reason why using stem case failed finding that relationship is that the word ‘havalandırma’ is generated from the stem ‘hava’.



CHAPTER FIVE

RESULTS

In this chapter of study, some statistical reviews of the software are represented. Over hundreds of use, all data, stop-words, relationships and etc. were stored and made up some statistical results for different purposes such as the percentage of the general success of the software, mostly used stop-words or the percentage of the most found relationship types.

To start with the general success of the study, the performance of the software has the rate of %78.87 success. The situation of finding at least one relationship from either case is accepted as the criteria of success. It is clear from the statistical ratios for the comparison of the cases that, the success of using stems are %77.46 while the success of using elements themselves are %65.72.

To talk about the relationships; in '*using stem*' case, same word relations are found as the most successful relationship type with the rate of %80.60.

Table 5.1 The success ratio of '*using stem*' relation types

	Same Word	Synonym	Antonym	Close-Meaning
%	80.60	75.15	9.09	3.03

In '*using elements themselves*' case, the previous order reappears. Same word relations are again calculated as the most successful relationship type with the ratio of %79.28.

Table 5.2 The success ratio of '*using elements themselves*' relation types

	Same Word	Synonym	Antonym	Close-Meaning
%	79.28	74.28	9.28	5.71

Stop-words are also used for gathering some statistical data. It is clearly understood from results that, the word ‘bir’ is the commonly used stop-word of Turkish language with the rate of %33.33.

- Bir %33.33
- Veya %27.61
- Ve %22.38
- Olan %15.23
- İçin %8.09
- Çok – her – kimse – olmayan %5.71

Aforementioned statistical results indicate that, using stems give better results compared to using elements themselves. However, even if ‘same word’ semantic relation has a higher ratio, the gap between ‘same word’ and ‘synonym’ relationships is not so wide. Antonym or close-meaning relationships indicate quite fewer clues compared to synonym or same word relationships.

Table 5.3 General view of statistic results of comparisons

WORD1	WORD2	Using Elements Themselves (UET)	Using Stems (US)	UET-SYN	UET-ACR	UET-CLOSE	UET-SAME	US-SYN	US-ACR	US-CLOSE	US-SAME
mıkyas	ölçek	2	4	1	0	0	1	2	0	0	2
kederlenmek	dertlenmek	2	2	2	0	0	0	2	0	0	0
öbürleri	diğerleri	2	1	1	0	0	1	1	0	0	0
kıyı	kenar	1	1	1	0	0	0	1	0	0	0
mir	alay	0	0	0	0	0	0	0	0	0	0
çokları	bazıları	0	0	0	0	0	0	0	0	0	0
haytalık	serseri	3	6	2	0	0	1	3	0	0	3
eksilmek	azalmak	3	2	1	0	0	2	1	0	0	1
sahileştirmek	gerçekleştirmek	4	3	2	0	0	2	1	0	0	2
karı	koca	5	9	3	0	0	2	5	0	0	4
komutan	asker	1	2	0	0	0	1	1	0	0	1
kaz	ördek	0	0	0	0	0	0	0	0	0	0
sanırmek	zannetmek	3	1	2	0	0	1	0	0	0	1
profesyonel	acemi	2	3	1	0	0	1	1	0	0	2
ajan	casus	4	2	2	0	1	1	1	0	0	1
arlanmaz	utanmaz	8	3	6	0	0	2	1	0	0	2
fazlalaşmak	azalmak	0	0	0	0	0	0	0	0	0	0
bilgilendirmek	bilgi vermek	0	0	0	0	0	0	0	0	0	0
çirkin	güzel	2	2	1	1	0	0	1	1	0	0
arz	talep	0	0	0	0	0	0	0	0	0	0
nazım	nazi	0	0	0	0	0	0	0	0	0	0
nazım	nazi	0	0	0	0	0	0	0	0	0	0
gerek	mecbur	0	0	0	0	0	0	0	0	0	0
itinalı	özenli	3	3	1	0	0	2	2	0	0	1
atmak	tutmak	0	0	0	0	0	0	0	0	0	0
acele etmek	çabuk	2	2	1	0	0	1	1	0	0	1
lokum	tatlı	1	1	1	0	0	0	0	0	0	1

CHAPTER SIX

CONCLUSION

Automatic Confirmation of Connotation Words is a software that determines the relationships between connotation words according to the result of a set of comparison operations. Using the system, the definition texts of the words are analyzed and cleared from stop-words which do not have specific meanings for their reference words. Following that, two separate element-lists are generated for each word. The system performs in the way of separating elements into stems or vice versa. the words in element-lists are replaced by their stems by using Zemberek software in using-stem case.

Comparisons of the words are performed by using ‘Turkish Synonym and Close-Meaning Dictionary’ and ‘Turkish Antonym Dictionary’, which both of them consist of more than 42000 word couples. Definition texts are taken from the list which consists of more than 115000 words with their lexical meanings.

Following the comparisons, several semantic relationships such as antonym, synonym, close-meaning and etc. of the words to be compared are found. The system can determine how strong semantic relationship those two words have according to the number of relationships. When system can not find any semantic relationship, those words are interpreted as not connotation words.

In every process, the system stores every information in order to maintain the capacity of making the statistical analysis such as the number of successful processes or number of semantic relation types or number of mostly used stop-words. In order to make computer understand or learn the semantics of words, those analyses are one of the major cores. Using stems, for example, instead of the elements themselves results better solutions. Moreover, having the same words/stems in their definition texts makes the semantic relationship stronger.

As other English language based algorithms can not be applied for Turkish, a sui generis version for Turkish language is needed. Generating the algorithm of automatic confirmation of connotation words will undoubtedly lead others to develop better ones. Besides, this study is a contribution to Turkish language researches in the way of dealing with the semantics of Turkish words.



REFERENCES

- Akın, A. A., & Akın, M. D. (2007). *Zemberek, an open source NLP framework for Turkic languages*. Retrieved December 09, 2015 from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.556.69&rep=rep1&type=pdf>.
- Aktaş, Ö. (2006). *Türkçe için verimli bir cümle sonu belirleme yöntemi*. Retrieved October 25, 2015 from <http://ab.org.tr/ab06/bildiri/213.pdf>.
- Aktaş, Ö., & Çebi, Y. (2013). *Rule-based sentence detection method (RBSDM) for Turkish*. Retrieved October 23, 2015 from <http://article.sciencepublishinggroup.com/pdf/10.11648.j.ijll.20130101.11.pdf>.
- Birant, Ç. C. (2008). *Root-suffix separation of Turkish words*. Master Thesis, Dokuz Eylül University, İzmir.
- Ekmekçioğlu, F., & Willett, P (2000). *Effectiveness of stemming for Turkish text retrieval*. Retrieved December 10, 2015 from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.2951&rep=rep1&type=pdf>.
- Eryiğit, G. (2012). *The impact of automatic morphological analysis & disambiguation on dependency parsing of Turkish*. Retrieved December 11, 2015 from http://www.lrec-conf.org/proceedings/lrec2012/pdf/198_Paper.pdf.
- Github (2015). *NZemberek kullanım*. Retrieved September 29, 2015 from <https://code.google.com/p/nzemberek/wiki/NZemberekKullanim>.

- Külekci, M., & Özkan, M. (2014). *Turkish word segmentation using morphological analyzer*. Retrieved December 12, 2015 from http://www.turkofoni.org/files/turkish_word_segmentation_using_morphological_analyzer_m._k_lek_.pdf.
- Lewis, M. P. (Ed.) (2009). *Ethnologue: languages of the world*. (16th Ed.). Dallas, Texas: SIL International.
- Nzemberek (2015). *Download zemberek library*. Retrieved September 29, 2015 from <https://code.google.com/p/nzemberek/downloads/list>.
- Orhan, Z., Pehlivan, İ., Uslan, V., & Önder, P. (2011). *Automated extraction of semantic word relations in Turkish lexicon*. Retrieved December 10, 2015 from https://www.researchgate.net/profile/Volkan_Uslan/publication/228530722_AUTOMATED_EXTRACTION_OF_SEMANTIC_WORD_RELATIONS_IN_TURKISH_LEXICON/links/0a85e53a2b2fde00d4000000.pdf.
- Sever, H., & Bitirim, Y. (2003). *FindStem: Analysis and evaluation of a Turkish stemming algorithm*. Retrieved December 11, 2015 from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.115.2511&rep=rep1&type=pdf>.
- Solak, A., & Oflazer, K. (1993). *Design and implementation of a spelling checker for Turkish*. Retrieved December 10, 2015 from <http://www.ceng.metu.edu.tr/~e124985/spellingchecker.pdf>.