BURAK ERTOPCU

M.S. Thesis

2017

# SENSE DISTINCTION USING COMPUTATIONAL METHODS IN TURKISH DICTIONARIES

BURAK ERTOPCU

IŞIK UNIVERSITY

2017

# SENSE DISTINCTION USING COMPUTATIONAL METHODS IN TURKISH DICTIONARIES

BURAK ERTOPCU
B.S., Management and Information Systems, IŞIK UNIVERSITY, 2014

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

IŞIK UNIVERSITY
2017

IŞIK UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

# SENSE DISTINCTION USING COMPUTATIONAL METHODS IN TURKISH DICTIONARIES

BURAK ERTOPCU

APPROVED BY:

| | | |
|---|---|---|
| Prof. Dr. Ercan Solak (Thesis Supervisor) | Işık University | _____ |
| Assoc. Prof. Olcay T. Yıldız | Işık University | _____ |
| Assoc. Prof. M. Oğuzhan Külekçi | Istanbul Technical University | _____ |

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

# SENSE DISTINCTION USING COMPUTATIONAL METHODS IN TURKISH DICTIONARIES

## Abstract

NLP(Natural Language Processing) refers to general name of the study fields related with processing languages by using computer-based systems. In NLP studies, dictionaries are required as lexical and semantic resources. Because in some cases, there are necessities to match the words with their correct senses for all possible words. There are some electronic dictionaries for Turkish such as "Contemporary Turkish Dictionary(CTD)" and "Kubbealtı Turkish Dictionary". However, both of these two dictionaries cover similar and redundant senses for several words.

There are 86.382 words exist in CTD that written by Turkish Linguistic Society(TDK). There can be more than ten senses for a single word in some cases. By that reason, it can be hard to determine which meanings are explanatory and/or required and which of them are multiplexed needlessly. This problem of finding distinguishing senses of the words is called as "Sense Distinction Problem". The aim of this study is to simplify the sense distinction decisions by using some computational methods.

In this study, we focused on to analyse the similarities of word senses by using some computational methods such as; Edit Distance, Cosine Similarity and Jaccard Index Similarity on two well-known Turkish Dictionaries Contemporary Turkish Dictionary (CTD) and Kubbealtı Dictionary (KD).

**Keywords: sense distinction, analysis of textual distance and similarity**

# TÜRKÇE SÖZLÜKLERDE HESAPLAMAYA DAYALI YÖNTEMLER İLE ANLAM AYRIMI

## Özet

Doğal Dil İşleme(NLP) herhangi bir dili bilgisayar bazlı sistemlerle işlemekle ilgili çalışma alanlarının genel ismidir. NLP çalışmalarında, sözcüksel ve anlamsal kaynaklar olarak sözlüklere ihtiyaç duyulmaktadır. Bunun sebebi, bazı durumlarda sözcük ile uygun anlamını eşleştirme gereksinimi bulunmasıdır. Türkçe için; "Güncel Türkçe Sözlük" ve "Kubbealtı Lugatı" gibi elektronik sözlükler bulunmaktadır. Ancak, bu iki sözlük de birçok sözcük için benzer ve çoklanmış sözcük anlamı içermektedir.

Türk Dil Kurumu(TDK)'nun Güncel Türkçe Sözlüğü 86.382 adet sözcük içermektedir. Tek bir sözcük için ondan fazla anlam karşılığı bulunabilir. Bu sebeple, hangi anlamların açıklayıcı ve/veya gerekli hangilerinin ise gereksizce çoklanmış olduğunu bulmak oldukça zorlaşabilir. Sözcüklerin anlamıyla ilgili yaşanan bu ayrıştırma problemine "Anlam Ayrımı Problemi" denir. Bu problem, NLP çalışmaları için minimal ve verimli bir sözlük üretmede önemli bir husustur. Özellikle Türkçe için, kelimelerin anlamları içerisinden en aydınlatıcı olanını seçmek pek kolay değildir. Bu çalışmanın amacı, anlam ayrımı kararlarını hesaplamaya dayalı bazı metodlar kullanarak kolaylaştırmaktır.

Biz bu çalışmada, en çok bilinen Türkçe Sözlük'lerden ikisinin(Kubbe Altı Lugatı ve TDK Güncel Türkçe Sözlük) üzerinde Levenshtein Mesafe Alogritması, Kosinüs Benzerliği ve Jaccard Benzerliği gibi hesaplamaya dayalı bazı metodlar kullanarak sözcük anlamlarının benzerliklerini analiz etmeye odaklandık.

**Anahtar kelimeler: anlam ayrımı, yazılar arası mesafe ve benzerlik**

## Acknowledgements

Special thanks to my supervisor Prof. Dr. Ercan Solak for sharing all of his background about the subject and for encouraging me to do research on it.

*To my family. . .*

# Table of Contents

# List of Tables

# List of Figures

# List of Abbreviations

| | |
|---|---|
| **TDK** | **T**ürk **D**il **K**urumu (Turkish Linguistic Society) |
| **KD** | **K**ubbe**A**ltı **L**ugatı (KubbeaAltı Dictionary) |
| **CTD** | **C**ontemporary **T**urkish **D**ictionary |
| **NLP** | **N**atural **L**anguage **P**rocessing |

# Chapter 1

# Introduction

There are several Turkish Dictionaries that are being used commonly for both academic purposes and also for daily communication. This study aims to analyse popular dictionaries of Turkish to determine the quality of sense distinctions in the definitions of a given lemma. Also, we aim to compare different dictionaries in terms of the similarities and the differences between their definitions and the number of distinct senses they identify. While doing these analysis, the ultimate purpose is to generate a minimal and efficient dictionary for providing a literal resource for NLP projects in Turkish. There are some problems about pre-processing stage. We tried to address orthographic ambiguities and other problems we encountered during textual comparisons related with word sense definitions.

There can be more than one meaning that may reach to large amounts per word in Turkish. In some cases, these different meanings are not so different in terms of usage. So, we need to handle with ambiguous definitions in comparison of these word sense definitions by using some text similarity calculation metrics.

As an example to explain this problem;

for the two different sense definitions of the word **"dün"(yesterday)** in Kubbealtı Dictionary (KD):

**1.** "İçinde bulunulan günden bir önceki gün" (the day before today)

**2.** "İçinde bulunulan günden bir önceki günde" (in the day before today)

Obviously, these two senses are very close. Thus, the natural question is if these two senses are actually distinct senses or just different usages.

After reviewing such cases, we decided to use some computational methods to measure these similarity ratios and also another main point was to see how different dictionaries differ in the number of senses they identify.

### 1.1 Reasons That Make Turkish Dictionaries Larger

The word counts in Turkish Dictionaries vary depending on scope of them. For instance, in "Ötüken" Turkish dictionary there are more than 200.000 words because the scope of this dictionary covers the ancient history of Turkish Language. In comparison, Contemporary Turkish Dictionary (CTD) of Türk Dil Kurumu (TDK) contains 86.382 words because this dictionary is based on modern Turkish words including some old words that are still in use. KD is also a modern Turkish dictionary that contains 44.247 words has similar features with CTD in terms of the age of the words.

Due to changes in Turkish Society's vocabulary set for centuries, there are new versions of old foreign (French, Arabic, Persian, English, Italian, and etc.) words arising. This issue increases the word frequency in Turkish dictionaries too. But this is not the only reason about the overgrowth of Dictionaries. When required words are determined for a dictionary, descriptive information for all these words are required too. The problem is; how to determine these definitions and the context of this definitions? This type of problems about word definitions are named as "Sense Distinction Problems". One of the important aspect that makes dictionaries bigger is insufficient sense distinction decisions.

In Turkish dictionaries, there are large amount of sense definitions for a single word and some of these definitions are completely similar in terms of semantics. This issue occurs because of several reasons (such as; multiple POS Tags for a

single word, metaphors and etc.). More detailed information about problematic issues about sense distinction for Turkish is in Section 1.2.

## 1.2 Sense Distinction Problems in Turkish Dictionaries

In Turkish Dictionaries most of all possible different senses for each word are covered. However, these different senses are not always different from each other in terms of usage. For example;

In GDT, the word "güzel"(beautiful, good, etc.) that is being used as both noun and adjective has these two meanings separately such as:

**1.** "İyi, hoş"(good, fine)

**example of usage:** "Güzel şey canım milletvekili olmak." (Being a parliamentarian is a good thing.)

**2.** "Sakin, hoş"(calm, fine)

**example of usage:** "Güzel bir gece" (It is a good night)

In the above example, there are two different sense definitions for a single word but these description texts identify the same meaning. The word "güzel" is being used for telling good/fine things in Turkish. So this issue of replication is not necessary for dictionaries. These problems exist in all other dictionaries that are written for Turkish. When facing with this ambiguous issue on dictionary building stage, picking one of these two similar sense definitions is an adequate approach.

In addition, within some cases in Turkish Dictionaries metaphors are being used as a new meaning of the same word. For instance;

In GDT, the word "yumuşak"(smooth, soft) that is used both as a noun and an adjective has these two meanings separately:

**1.** "Dokunulduğunda veya üzerine basıldığında çukurlaşan, eski biçimini kaybeden, katı karşıtı." (The thing that looses its old shape or becomes hollow when it had been touched or it had been pressed, the opposite of solid.)

**example of usage :** "Pamuk yumuşaktır." (Cotton is soft.)

**2.** "Kaba, hırçın, sert olmayan, kolay yola gelen, uysal." (The person who is not impolite, combative and etc.)

There is no example of usage for this sense definition in GDT.

In the above example, selecting one of them as the correct meaning of "yumuşak"(soft) is not sufficient because both of them are necessary. One of them defines the real meaning and the other one is a metaphor with the same word "yumuşak". Both of these two sense definitions are required for this lemma.

These type of issues that require a certain decision about picking the reasonable meanings of the words are called as "Sense Distinction Problems" in dictionaries. In order to handle with these problems, qualified identification of word senses and also usage examples per each word sense is required in dictionaries.

# Chapter 2

# Literature Survey

Sense Distinction is a requirement for studies about dictionaries. There is not any study about sense distinction using computational methods in Turkish Dictionaries. So in general, related studies for English dictionaries are surveyed within this project.

The study [1] is focused on Word Sense Distinctions in English and some other European Languages. In this study, the author focuses on problems about obtaining word sense identifications. In addition, he claims that, lexicographers usually do not write about how to write a dictionary. He determines that the central task for writing a dictionary is to specify the word meanings. In conclusion, he states that without any clear definitions of word usages, word senses are not informative, because the context of sentences affect the word senses directly.

In study [2], they focus on sense distinction using annotations for word senses in English. The annotations considered are both automatically done by the system that is built by using machine learning techniques and manually done by humans. They found different types of errors in results for both the automatic and manual annotations. The causes of errors are categorized as; insufficient sense entries, uncertain context, and insufficient word knowledge. The authors state that their results based on computational methods are sufficient for practical usage of words with their convenient sense definitions.

In the article [3] he describes the evolution of lexicography that starts with corpus-driven studies. He states that for modern lexicography, intuitions of native speakers are not effective for matching the words with their appropriate senses within a given text any more. He states that inadequate word sense definitions in dictionaries are based on several errors about the practical usage of the words. He describes that in the future of lexicography, there will be several attempts to create electronic tools that can relate current usage of words in texts with their convenient meanings.

The study [4] aims to discover word senses from texts. They prefer to cluster the words hierarchically. They use corpus-driven data. Their word sense extraction algorithm is determined as learning based. They use cosine similarity for finding similar words. They find clusters of words by applying recursive methods within the similarity space. They assign each word to clusters. In the results of this study, their algorithm finds 2869 polysemous words in total of 13.403 words. Their results for their algorithm in F-score is 63.1% on the test set with a specific learning rate. Their results indicate that the cosine similarity based clustering method is accurate than the others such as K-means.

# Chapter 3

# Data

We already have CTD data in a structured form. All the data of this project is in textual format. We stored them in JSON files. All words are JSON objects and they have several attributes.

Our structured format is shown in the example below for the word **"düşürmek" (cause to fall):**


"düşürmek": [
{
"alternation": "",
"domain": [],
"literal": "düşürmek",
"origin": "",
"pos": ["oldurgan fiil"],
"pronunciation": "",
"senses": [
{
"definition": "Düşmesine yol açmak, düşmesine sebep olmak",
"domain": [],
"example": "Ben şimdi buracıkta tarağımı düşürmüşüm, gördünüz mü?",
"pos": []
},

{
“definition”: “Değerini, fiyatını indirmek”,
“domain”: [],
“example”: null,
“pos”: []
},
{
“definition”: “Azaltmak”,
“domain”: [],
“example”: null,
“pos”: []
},
{
“definition”: “Vücuttan yavru, çocuk, taş, solucan vb. atmak”,
“domain”: [],
“example”: “çocuk solucan düşürüyor.”,
“pos”: [“nsz”]
},
{
“definition”: “Görevi bıraktırmak”,
“domain”: [],
“example”: “Bakanlar kurulunu düşürmek.”,
“pos”: []
},
{
“definition”: “Uğratmak”,
“domain”: [],
“example”: “Tehlikeye düşürmek.”,
“pos”: []
},
{

“definition”: “Değerli bir şeyi ucuz veya kolay elde etmek”,
“domain”: [],
“example”: null,
“pos”: []
},
{
“definition”: “Zayıf bırakmak, gücünü azaltmak”,
“domain”: [],
“example”: “Annemi verem iyiden iyiye düşürmüştü.”,
“pos”: []
}
],
”usage”: []
}
]

In order to build this structure, we handled some problems that occur in the raw data which is in HTML format. Details of these problems described in Section 3.2.

Also, we need a second dictionary for comparison. We selected the KD as the second one for our comparisons. However, we faced with some problems while collecting the dictionary data of KD. The details about these problems are described in Section 3.1.

In total, we had 44.761 words from KD dictionary, and 86.382 words from CTD. All of these words are used as word objects on analysis.

### 3.1 Problem About Collecting Data

We collected our data from the official websites for both CTD and KD by using Python scripts. In collection stage of KD, we faced with a problem. The problem was to collect all words that exist in KD with a single query. This issue required a solution to find a parameter during the search of all words in a single query.

In order to fix this issue, we found the solution as sending ampersand symbol("&amp") literally to get all words by sending in just one query. By that solution, we collected all words and their sense definitions that are given in pagination. In order to parse these raw data into our standard JSON dictionary format, we applied some pre-processing.

### 3.2 Data Conversion Process

To transform the raw data into a convenient dataset, we converted HTML files into JSON objects per each word. In this stage, we faced with some problems about KD data.

Here is a sample about these raw HTML files for the word "güzel"(beautiful, good, fine):

---

```
<div id="content">
<div class="search">
<div class="keyPad"></div>
<div class="searchBox">
<div class="type-helper">
<div class="tooltipDiv"><span
    style="float:left;display:block;width:40px;"><h2> </h2></span>
```

```
<div class="search-results-div"><h3>GZEL</h3><div
    class="search-results"><p class="quota"><span
    class="ChampturkI150">sf. </span><span class="Champturk14">(&lt;
    </span><span class="ChampturkI14">gzel </span><span
    class="Champturk14">&lt; </span><span
    class="ChampturkI14">gz+el</span><span class="Champturk14">)
    </span><br/>
<span class="ChampturkB150">1. </span><span class="Champturk150">Gze
    ve kulaa ho gelen, grlmesi, duyulmas insana zevk veren, cemil.
     Kart : </span><span class="Champturk14"> RKN : </span><span
    class="ChampturkI150">Gzel yz . Gzel vcut . Gzel
    manzara. Gzel kalarnn arasnda incecik bir izgi belirmiti
    </span><span class="Champturk150">(Trk Bura ). </span><br/><span
    class="ChampturkB150">2. </span><span class="Champturk150"> Tad
      stn nitelikler veya istee uygun olmas sebebiyle insanda iyi
    etki brakan, takdir uyandran: </span>
<span class="ChampturkI150"> Gzel huy. Gzel fikir. Gzel
    haber. Gzel duygu. Gzel yolculuk. Gzel
       gr . Sflerden biri, Allahn emrettii ey gzel ,
      yasaklad ey irkindir demitir </span><span
    class="Champturk150">(Taarruf Terc.). </span><span
    class="ChampturkI150">imde dalgal tekbri en gzel dnin </span><span
    class="Champturk150">(Yahy Kemal). </span><br/><span
    class="ChampturkB150">3. </span><span class="Champturk150">(Hava
    iin) Ak, skin, ho: </span><span class="ChampturkI150">Bugn hava
    gzel </span><span class="Champturk150">(Chit S. Taranc).
    </span><span class="ChampturkI14">i. </span><br/><span
    class="ChampturkB150">4. </span>
```

ekici, czip kimse [zellikle gen kz ve kadnlar hakknda kullanlr ]: Gzelsiz yaylaya konup glmez (Karacaolan). Bir gzel sevds serimde tter (Pir Sultan Abdal). Grdn her bir cemli hsn-i Ysuf sanma kim / ok gzeller vardr amm hsn -i Kenan bir olur (Nreddin Kalkandelen). <br/>5. 

nsanda hoa giden bir etki ve estetik duygu uyandran eyin nitelii: Bu nokta insan olunun iyiye, gzele olan kbiliyetlerinden baka ne olabilir? (Ahmet H. Tanpnar). <br/>6. (sim tamlamasnn ikinci esi olarak) Belli bir evre iinden aranlan niteliklere en uygun olarak seilen kimse: Trkiye gzeli. Plaj gzeli. Tekirdada her sene karpuz gzeli seilir . <br/>7. 

zf. yi, ho, l: Hi ummadn yerde / Ngh alr perde / Derman eriir derde / Mevl grelim neyler / Neylerse gzel eyler (Erzurumlu brhim Hakkdan ). Bitmi veya tam diyebileceimiz hibir eser bu topran mcersn bu kadar gzel hulsa edemez 

(Ahmet H. Tanpnar). Hiciv ve methiye kark bu beyitlerde Trk rhunun neesi o kadar gzel gzkyordu ki! (Ren E. naydn ). <br/>8. nl. 

Pekiyi, pekl, doru : Kaa aldn ? 10 milyon. Gzel ! Bileti aldn , gzel , gzel ama ya otelde yer bulamazsak.  <br/>

 Gzel gzel: stenene uygun ekilde, arzu edildii gibi, uslnce: Gzel gzel otur. Gzel gzel ye. Gzel olmak: 

Gzel bir hal almak, istee uygun bir ekil almak, istenildii gibi olmak: Limon koyunca orba gzel oldu.  <br/>Gzelce zf. Gerektii gibi, adamakll, iyice: Gzelce temizle, gzelce yka. Benim kzm hamarattr, gzelce hizmet eder (Hseyin Sret).

```
</span><br/><span class="ChampturkB16"> Gzelim </span><span
class="ChampturkI150">sf. </span><span
class="Champturk150">(</span><span class="Champturk14">yelik ekinin
kalplamasyle </span><span class="ChampturkI14">gzel+i+m</span><span
class="Champturk150">) </span><br/><span class="ChampturkB150">1.
</span><span class="Champturk150">Pek gzel olan, beenilip hoa
gitmekte olan, gzel olup sevilen: </span><span
class="ChampturkI150">Gzelim kza yazk oldu. te kimse u gzelim
hayvana fiyatn vermemiti </span><span class="Champturk150">(mer
Seyfeddin). </span><span class="ChampturkI150">Gzelim bahar rzgrnda
ter kokular </span><span class="Champturk150">(Orhan V. Kank).
</span><br/><span class="ChampturkB150">2. </span><span
class="ChampturkI14">nl. </span><span
class="Champturk150">Beenilen, sevilen kimselere hitap sz olarak
kullanlr, ekerim: </span><span class="ChampturkI150">Gzelim, biraz
bakar msn? </span><span class="Champturk150">
</span></p></div></div> <br/><div
class="search-results-div"><h3>GZEL AVRAT OTU</h3><div
class="search-results"><p class="quota"><span
class="ChampturkI150">birl. i. </span><span
class="Champturk150">Patlcangillerden, scak ve lk blgelerde yetien,
mor iekli, siyah ve tatlms, kiraza benzer bir meyvesi olan, pis
kokulu, atropin denen zehiri tad iin tpta kullanlan otsu bitki.
</span><span class="ChampturkB14">Atropa belladonna. </span><span
class="Champturk150"> </span></p></div></div> <br/><div
class="search-results-div"><h3>GZEL HATUN E</h3><div
class="search-results"><p class="quota"><span
class="ChampturkI150">birl. i. </span><span
class="Champturk150">Soanla retilen, iri ve gzel pembe iekli bir ss
bitkisi, nergis zamba. </span><span class="ChampturkB14">Amaryllis
belladonna. </span><span class="Champturk150">
</span></p></div></div> <br/><div
class="search-results-div"><h3>GZEL SANATLAR</h3><div
class="search-results"><p class="quota"><span
class="ChampturkI150">birl. i. </span><span
class="Champturk150">Resim, heykel, mmr, tezhip, hat vb. el
sanatlarna ve temil yoluyle msik, tiyatro, edebiyat gibi sanatlara
```

```
</span></p></div></div> <br/><div
    class="search-results-div"><h3>GZELLEME</h3><div
    class="search-results"><p class="quota"><span
    class="ChampturkI150">i. </span><span class="Champturk14">(&lt;
    </span><span class="ChampturkI14">gzel+le-me</span><span
    class="Champturk14">) </span><br/><span class="ChampturkB150">1.
    </span><span class="Champturk150">Halk edebiytnda sevgilinin veya
    bir yerin gzelliini vmek iin yazlan koma tarznda iir.
    </span><br/><span class="ChampturkB150">2. </span><span
    class="Champturk150">Bu tarz iirlerin bestelenmesinden meydana
    gelen halk msiksi tr.
</span></p></div></div> <br/><div
    class="search-results-div"><h3>GZELLENMEK</h3><div
    class="search-results"><p class="quota"><span
    class="ChampturkI150">geisiz f. </span><span
    class="Champturk14">(&lt; </span><span
    class="ChampturkI14">gzel+len-mek</span><span class="Champturk14">)
    </span><span class="ChampturkBI150">E. T. Trk. ve halk az.
    </span><span class="Champturk150">Gzel olmak, gzellemek:
    </span><span class="ChampturkI150">plp sevilen yr gzellenir
    </span><span class="Champturk150">(Karacaolan). </span><span
    class="ChampturkI150">Gzellendi havlar evvel-i fasl- zemistandr /
    arb i gl gibi anma bahr- lem-ry </span><span
    class="Champturk150">(Rh-i Badd). </span><span
    class="ChampturkI150">Hayt- tze buldu yine lem nev-bahr oldu /
    Gzellendi emen bir lle -hadd gl - izr oldu </span><span
    class="Champturk150">( eyhlislm Yahy ).
```

```
</span></p></div></div> <br/><div
    class="search-results-div"><h3>GZELLEMEK</h3><div
    class="search-results"><p class="quota"><span
    class="ChampturkI150">geisiz f. </span><span
    class="Champturk14">(&lt; </span><span
    class="ChampturkI14">gzel+le-mek</span><span class="Champturk14">)
    </span><span class="Champturk150">Gzel olmak, gzel duruma gelmek:
    </span><span class="ChampturkI150">Her sene biraz daha gzelleecek
    bahar </span><span class="Champturk150">(Ziy O. Sab). </span><span
    class="ChampturkI150">tiraf edeyim ki Nln hibir gn bu derece
    gzellememiti </span><span class="Champturk150">(Kerme Ndir).
</span></p></div></div> <br/><div
    class="search-results-div"><h3>GZELLETRLMEK</h3><div
    class="search-results"><p class="quota"><span
    class="ChampturkI150">edilgen f. </span><span
    class="Champturk14">(&lt; </span><span
    class="ChampturkI14">gzelletir-i-l-mek</span><span
    class="Champturk14">) </span><span class="Champturk150">Gzel duruma
    getirilmek.
</span></p></div></div> <br/><div
    class="search-results-div"><h3>GZELLETRMEK</h3><div
    class="search-results"><p class="quota"><span
    class="ChampturkI150">oldurgan f. </span><span
    class="Champturk14">(&lt; </span><span
    class="ChampturkI14">gzelle-tir-mek</span><span
    class="Champturk14">) </span><span class="Champturk150">Gzel duruma
    getirmek, gzellik vermek: </span><span class="ChampturkI150">nce
    profili, mehtbn her eyi gzelletiren bys iinde ktan bir heykel
    </span><span class="Champturk150">(Yusuf Z. Orta).
```

```
</span></p></div></div> <br/><div
    class="search-results-div"><h3>GZELLK</h3><div
    class="search-results"><p class="quota"><span
    class="ChampturkI150">i. </span><br/><span class="ChampturkB150">1.
    </span><span class="Champturk150">Gzel olan eyin nitelii, gze,
    kulaa ho gelen veya tad stn niteliklerle insanda iyi etki brakan,
    takdir uyandran ey veya hlin durumu, hsn, cemal, behet:
    </span><span class="ChampturkI150">Yz  gzellii .  Huy
     gzellii .  Karadr  kalar  benzer  kmre  / Bu gzellik ziyan
    verir  mre  </span><span class="Champturk150">(Trk). </span><span
    class="ChampturkI150">iirler kemer ve stunlarn  gzelliini
     sylyordu  </span><span class="Champturk150">(Ahmet Him).
    </span><span class="ChampturkI150">Bu gzelliinle sen / Bir sihirli
    gnesin </span><span class="Champturk150">(Orhan S. Orhon).
    </span><br/><span class="ChampturkB150">2. </span><span
    class="Champturk150">Yumuak, tatl sz veya davran: </span><span
    class="ChampturkI150">O kadar kibar davranma, gzellikten anlamaz.
    unu gzellikle sylesen olmaz m?  </span><br/><span
    class="ChampturkB150">3. </span><span class="ChampturkBI14">eski.
    </span><span class="Champturk150">Gzellemek iin yze srlen dzgn.
</span></p></div></div> <br/>
</div>
</div>
</body>
```

The first issue was splitting the data into lemmas. In order to split the data into lemmas we used "<h3>" tags. Because all of the lemmas were identified with this tag.

However, in some cases there are multiple lemmas that have different meanings. These lemmas are not duplicates, they are homonyms. For instance the word "aksak"(lame) has two entries. So we get these entries separately. We handled

these issues by using object list structure for each word and if there are more than one entries, we split them by using "<h3>" tag as a new word with checking if the existing word is the same with previous one. If this control mechanism returns "true" we appended the existing entry to the object list of the specified word.

After splitting the data into words that cover all homonym entries, we split all senses of these words if the specified word has more than one senses. In order to perform this operation, we found a special character that is called as "Broad on" which is a Cyrillic Letter that tells us the division of senses is starting. However, in some cases this character was replaced with a specific HTML tag "". By checking both this character and tag we found the starting point of the sense division. The ending point was the next <h3>tag that was being used to obtain the starting point of a new word. By extracting this information, we determined the word sense division boundaries.

By using these boundaries we distinct each sense definition texts for each word if the word has more than one meanings. In order to split the senses, we found that all senses start with numeric characters. These numbers give the sense order as expected. By using this numeric characters, we split the sense definition texts per each word.

# Chapter 4

# Experiments

We used calculation based comparative approach for finding the distances/similarities between sense definitions of words. In this study, we implemented three generic text comparison methods; which are "Edit Distance", "Cosine Similarity" and "Jaccard Similarity Index".

In addition, we designed a visualization tool to check orthography for sense definition similarities. It covers the entry size and sense definition count for each word and shows all of the word senses for a given word. Details about this tool are in Section 4.7.

## 4.1 Implementation of Similarity Analysis

In order to apply text comparison metrics we designed an analysis tool. This tool is designed as a console application for providing better performance on runtime.

To prepare the data for our calculations we decided to apply a pre-processing stage on our dictionary dataset. For instance, to apply Cosine Similarity and Edit Distance, we converted textual word sense definitions into row vectors. To vectorize these textual data we had used "TextToVec" approach. We also transformed these data into set to apply Jaccard Index Similarity. Details about our pre-processing are in Section 4.2.

We implemented the distance/similarity metrics separately as functions. The details about "Edit Distance" are in Section 4.4, and for "Cosine Similarity" the details are in Section 4.5, the details about "Jaccard Similarity Index" are in 4.6. Also the details of "Dictionary Visualization Tool" are described in 4.7, and the details of "TextToVec" approach are described in Section 4.3.

## 4.2 Pre-processing Stage

For preparing the data for calculations, we ignored all of the punctuations but they were not removed for using them to provide data clarification. In some cases, we used dot(.) character to extract the POS (Part-of-Speech) tag from the sense definition text.

We mapped some special Turkish characters into Latin Alphabet characters, such as; "â" to "a" for both lemmas and senses to prevent confusions related with orthography. We also ignored the empty sense definition texts. For each distance/similarity metric we defined the functions that implement the formulas of these metrics.

## 4.3 TextToVec Approach

We used a text vectorization method that is based on characters not words. We indexed all of the characters for each input text with integers. By that way, we provided for row vectors of word senses to have equal length in comparisons. These vectors contain the count value for each letter. This vectorization approach used to identify the similarity between two words that one of them is in form "root+suffix" and the other one is just in the "root" form for the same root.

For example;

for "input a" as **"araba"(the car)** and "input b" as **"arabacı"(the driver of a car)**:

A = [“a”:3, “b”:1, “c”:0, “d”:0, ....., “z”:0 ]

B = [“a”:3, “b”:1, “c”:1, “d”:0, .....,“ı”:1....., “z”:0 ]

If the text vectorization approach is based on words, these two words are completely different. We prevent this issue because these words are directly relational in terms of usage and semantics. By that reason, we used character-based Text-ToVec Approach.

### 4.4 Edit Distance

The “Edit Distance” Metric returns a positive integer value that identifies the distance between two texts. We used Levenshtein’s Algorithm to implement the “Edit Distance”. This method have three operations that are; “insert”, “update”, and “delete”. By using them it determines how much operations is required to get target string from the source string. The required operation count is the result of Edit Distance.

As an example;

from “can”(life,soul) to “kan”(blood)

the only operation needed is to update the starting letter “c” to “k”. By that reason the edit distance value for this example is 1.

When the source and target strings have the same short length, their resulting Edit Distance value is small. But in some cases, there are larger distances between the source and the target.

Such an example, for two different meanings of the Turkish word **“kavurma”**(braised meat, and in Turkish homonym of roasting),

**1.** “Kendi yağı ile kavrulduktan sonra ileride yenmek üzere saklanan et”(Meat that is stored for eating after roasting with its own oil),

**2.** "Kavurmak işi"(roasting),

for this case, the Edit Distance value between these two sense definitions is 58.

The Edit Distance results vary depending on both the length and the differences of characters between the source and the target. By that reason we calculated the Edit Distance values for each word sense pairs of all words.

### 4.5 Cosine Similarity

Another metric that we used is "Cosine Similarity". It returns a real number between 0 and 1 as the similarity ratio between two texts. To use this metric, we vectorize all of the sense definitions for each lemma. "Cosine Similarity" takes two inputs of integer vectors that represents input text A and B contain count values for each letter of these two inputs. In this study, the first index of this vector contains the count value for letter "a", and the rest are similarly calculated for each letter in Turkish.

As an example: For the word "adil"(fair)

V = ["a":1, "b":0, "c":0, "d":1, ..... , "z":0]

The calculation for obtaining Cosine Similarity is done for vector "A" and vector "B" by using the formula below;

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

### 4.6 Jaccard Similarity Index

In this study, we also used Jaccard Similarity Index to calculate the ratio of similarity between senses of words within the dictionaries. This method is one of the alternative ways to obtain the similarity ratio (between 0 and 1) for two textual inputs. We defined sets that contain count values for input texts for each

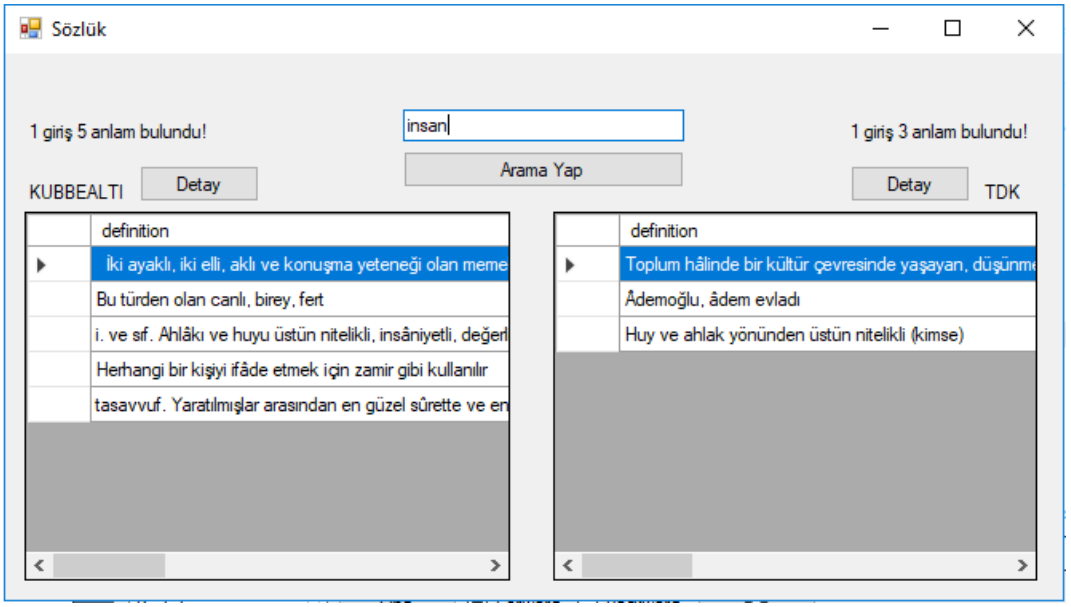


Figure 4.1: Sample run for the noun n as "insan"(human).

sense pair. We implemented the formula of similarity ratio based on this metric as below;

$$\text{JaccardIndex(A,B)} = \frac{|(A \cap B)|}{|(A \cup B)|}$$

## 4.7 Dictionary Visualization Tool

To check the form of the data and to explore the resulted similarity ratios between word senses visually, we designed a dictionary comparison software by using C# as the programming language. It binds all the words and all properties of each word objects coming from both CTD and KD. It returns all properties of word that is typed by the user.

In Figure 4.1 a sample run of this software for the noun "insan"(human) is shown.

In Figure 4.2 a sample run of this software for showing the properties of the noun "insan"(human) is shown.

In Figure 4.3 a sample run of this software for the verb "yardım etmek"(to help) is shown.

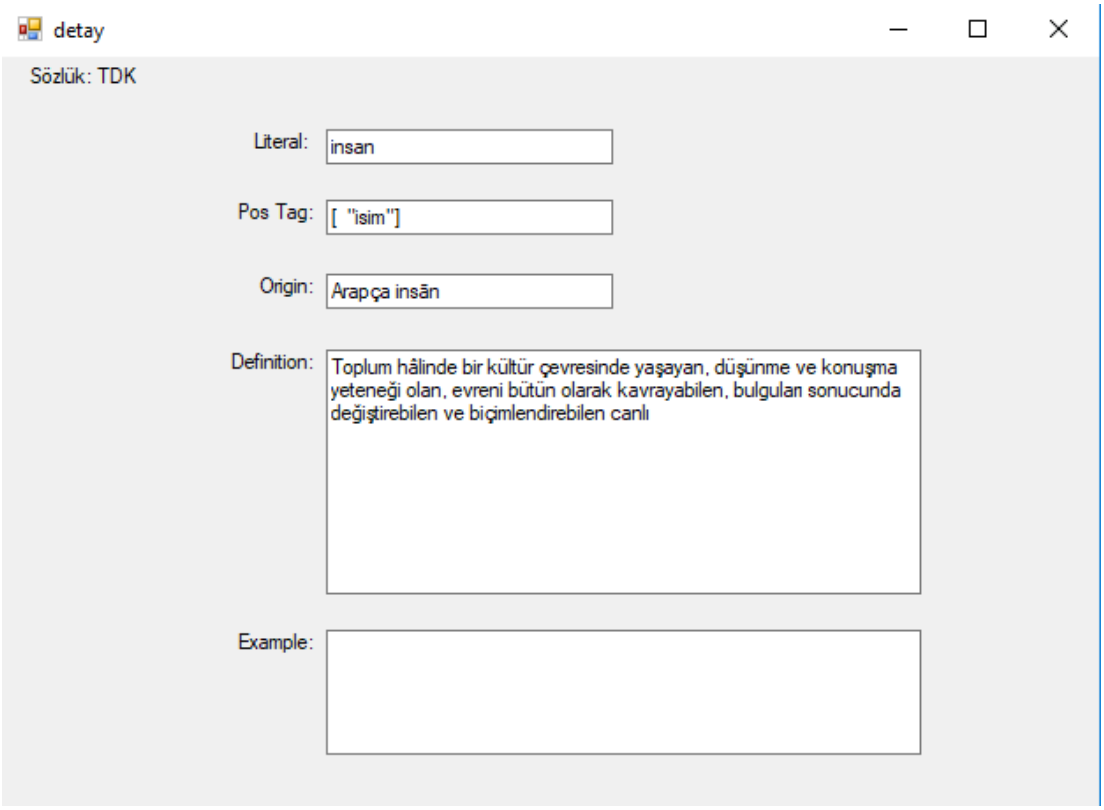


Figure 4.2: Sample run for the details of noun n as “insan”(human).

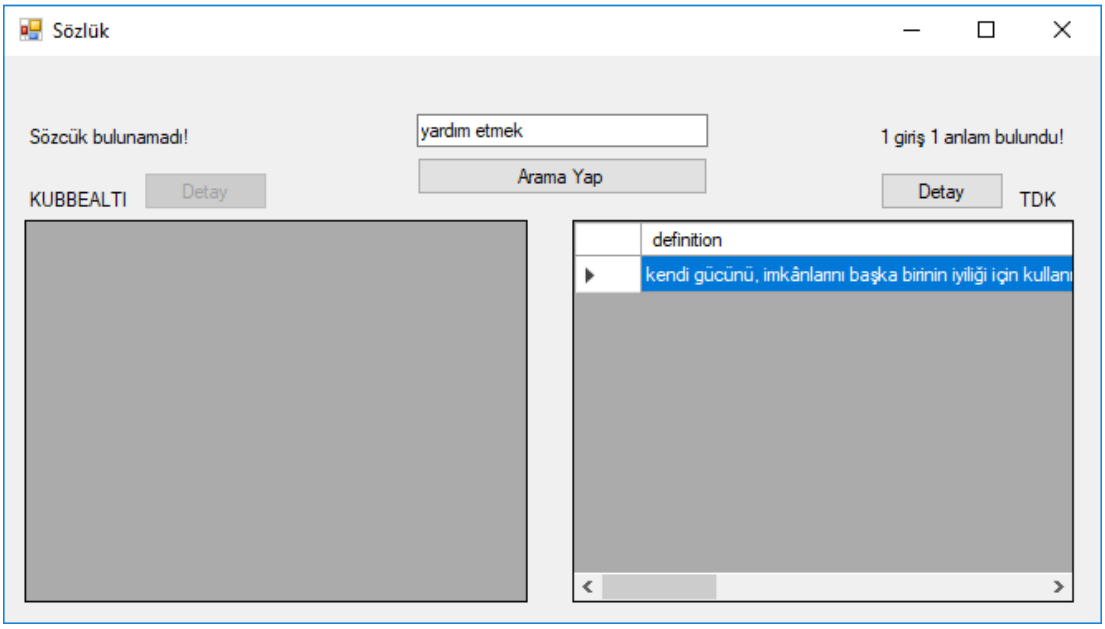


Figure 4.3: Sample run for the verb v as “yardım etmek”(to help).

# Chapter 5

# Results and Discussion

We divided our results into two main pieces as; results that cover word sense comparison of two Dictionaries for each word and results that cover word sense analysis per dictionary on its own for each word. Our data consist of entries per word and definitions per entry.

Table 5.1 shows the resulting analysis about the entry size and total definition count for sample of 5 words in KD:

Table 5.2 indicates the resulting analysis about the entry size and total definition count for sample of same 5 words in CTD:

Table 5.1: Entry and Definition Size Sample for KD

| **Lemma** | **Entry Size** | **Total Definition Size** |
|---|---|---|
| akrepler | 1 | 1 |
| akrilik | 1 | 2 |
| akrobasi | 1 | 2 |
| akrobat | 1 | 1 |
| akrobatik | 1 | 1 |

Table 5.2: Entry and Definition Size Sample for CTD

| **Lemma** | **Entry Size** | **Total Definition Size** |
|---|---|---|
| akrepler | 1 | 1 |
| akrilik | 1 | 2 |
| akrobasi | 1 | 1 |
| akrobat | 1 | 1 |
| akrobatik | 1 | 1 |

Table 5.3: Similarity Results between CTD and KD for 5 sample words

| **Lemma** | **Edit Distance** | **Cosine Sim.** | **Jaccard** | **Sense Pair** |
|---|---|---|---|---|
| akrepler | 0.44 | 0.92 | 0.18 | [(0, 0), (0, 0)] |
| akrilik | 0.69 | 0.88 | 0.27 | [(0, 0), (0, 0)] |
| akrobasi | 0.25 | 0.75 | 0.20 | [(0, 0), (0, 0)] |
| akrobat | 0.10 | 0.32 | 0.08 | [(0, 0), (0, 0)] |
| akrobatik | 0.31 | 0.80 | 0.10 | [(0, 0), (0, 0)] |

Table 5.4: Similarities for 5 sample words inside CTD

| **Lemma** | **Edit Distance** | **Cosine Sim.** | **Sense Pair** |
|---|---|---|---|
| akrepler | has only 1 sense | | |
| akrilik | 0.33 | 0.64 | [(0, 1), (0, 0)] |
| akrobasi | has only 1 sense | | |
| akrobat | has only 1 sense | | |
| akrobatik | has only 1 sense | | |

## 5.1 Similarity Ratios Between Dictionaries in Terms of Word Senses

Table 5.3 indicates the output of Edit Distance, Cosine Similarity and also Jaccard Similarity Index of word senses for 5 sample words between two dictionaries with word sense indexes. (Leftmost indexes are for KD and rightmost ones are for CTD.) The meaning of these indexes are; [($i^{\text{th}}$ entry of word w in KD, $j^{\text{th}}$ sense of w in entry), ($k^{\text{th}}$ entry of word w in CTD, $l^{\text{th}}$ sense of w in entry)]:

## 5.2 Similarities Between Word Senses inside Each Dictionaries Separately

Table 5.4 defines the output of Edit Distance, and Cosine Similarity in terms of word senses for 5 sample words inside a single dictionary(CTD). Indexes are designed as: [($i^{\text{th}}$ entry of word w in CTD, $j^{\text{th}}$ sense of w in entry), ($k^{\text{th}}$ entry of word w in CTD, $l^{\text{th}}$ sense of w in entry)]:

Table 5.5 indicates the output of Edit Distance, and Cosine Similarity in terms of word senses for 5 sample words inside a single dictionary(KD). Indexes are

Table 5.5: Similarities for 5 sample words inside KD

| Lemma | Edit Distance | Cosine Sim. | Sense Pair |
|---|---|---|---|
| akrepler | has only 1 sense | | |
| akrilik | 0.30 | 0.82 | [(0, 1), (0, 0)] |
| akrobasi | 0.41 | 0.84 | [(0, 1), (0, 0)] |
| akrobat | has only 1 sense | | |
| akrobatik | has only 1 sense | | |

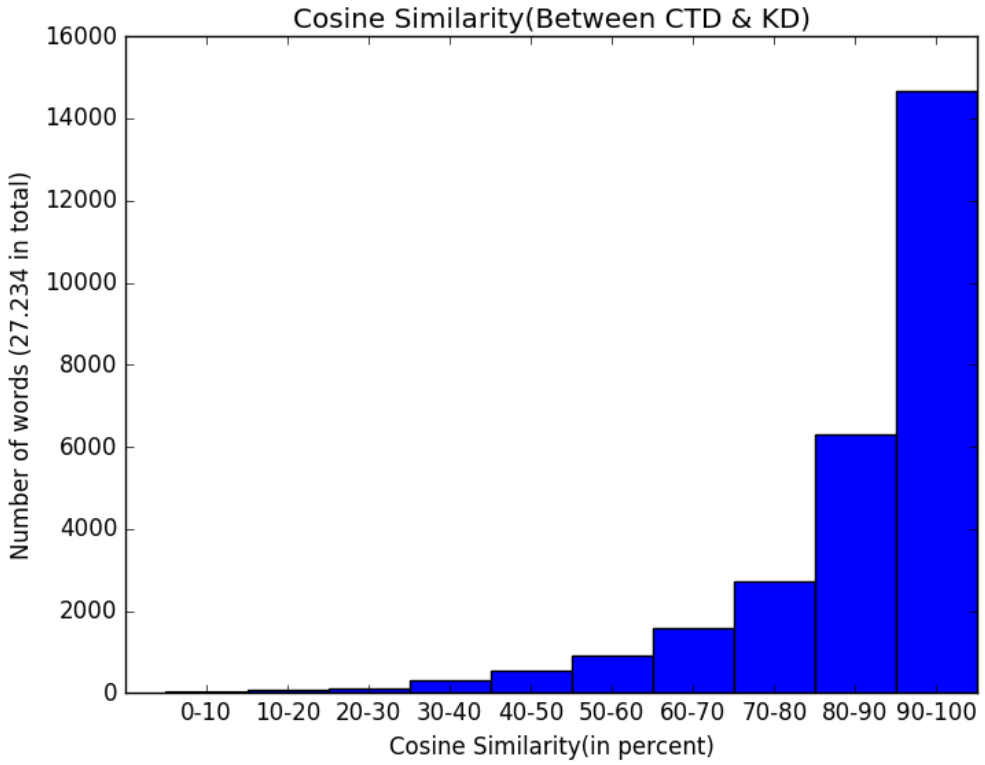


Figure 5.1: Distribution of similarities for word senses between CTD and KD by using Cosine Similarity.

designed as: [($i^{\text{th}}$ entry of word w in KD, $j^{\text{th}}$ sense of w in entry), ($k^{\text{th}}$ entry of word w in KD, $l^{\text{th}}$ sense of w in entry)]:

In Figure 5.1, the Cosine Similarity analysis is based on maximum ratios per word within sense pairs between CTD and KD.

In Figure 5.2, the Edit Distance analysis is based on maximum ratios per word within sense pairs between CTD and KD.

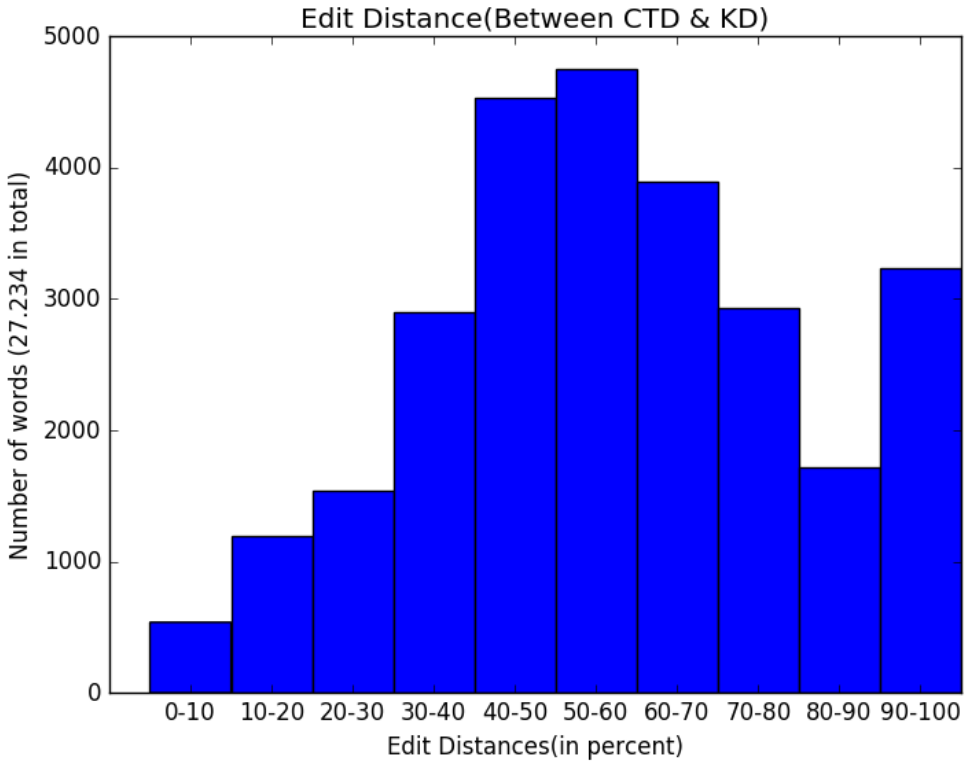


Figure 5.2: Distribution of similarities for word senses between CTD and KD by using Edit Distance.

In Figure 5.3, the Edit Distance analysis is based on maximum ratios per word within sense pairs inside CTD.

In Figure 5.4, the Edit Distance analysis is based on maximum ratios per word within sense pairs inside KD.

In Figure 5.5, the Cosine Similarity analysis is based on maximum ratios per word within sense pairs inside CTD.

In Figure 5.6, the Cosine Similarity analysis is based on maximum ratios per word within sense pairs inside KD.

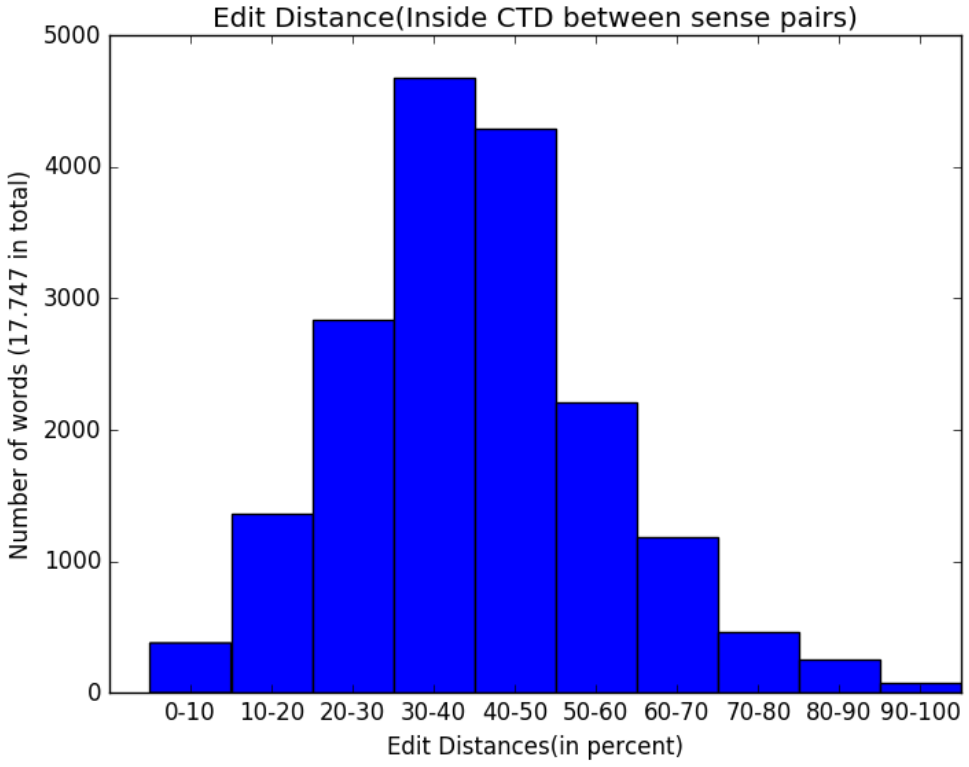


Figure 5.3: Distribution of similarities for word senses inside CTD by using Edit Distance.

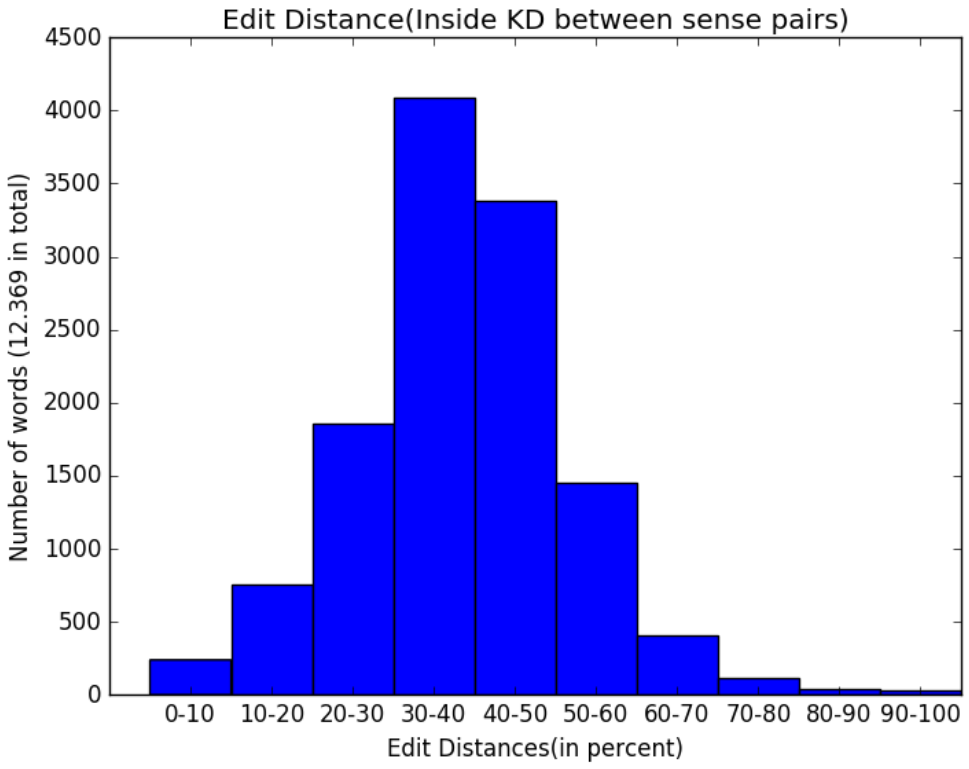


Figure 5.4: Distribution of similarities for word senses inside KD by using Edit Distance.

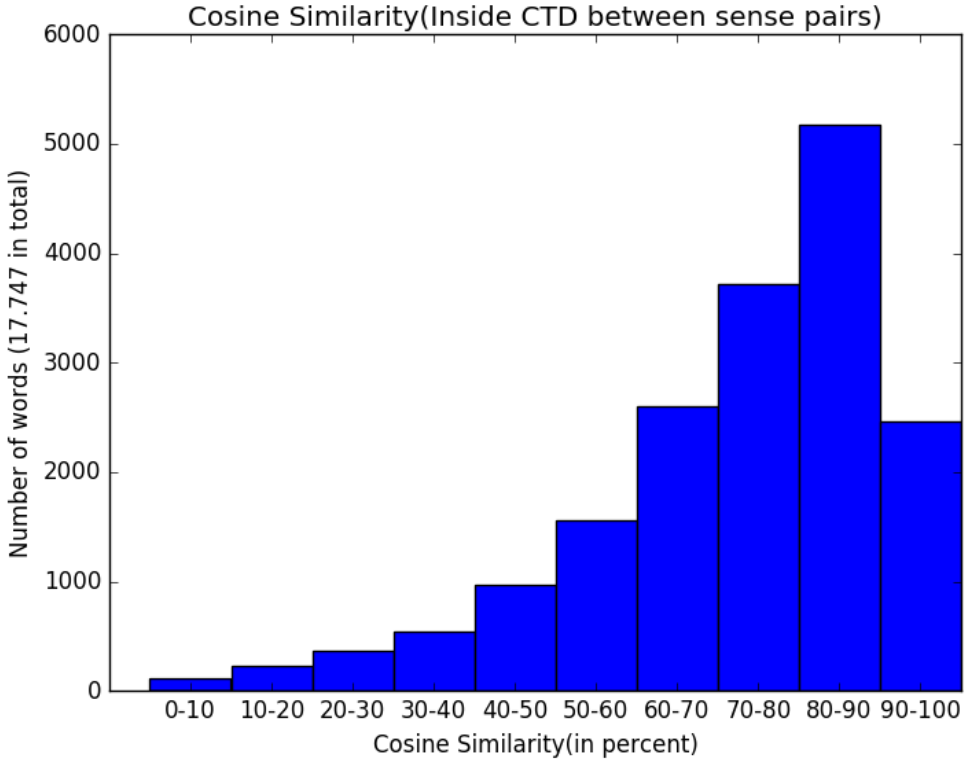


Figure 5.5: Distribution of similarities for word senses inside CTD by using Cosine Similarity.

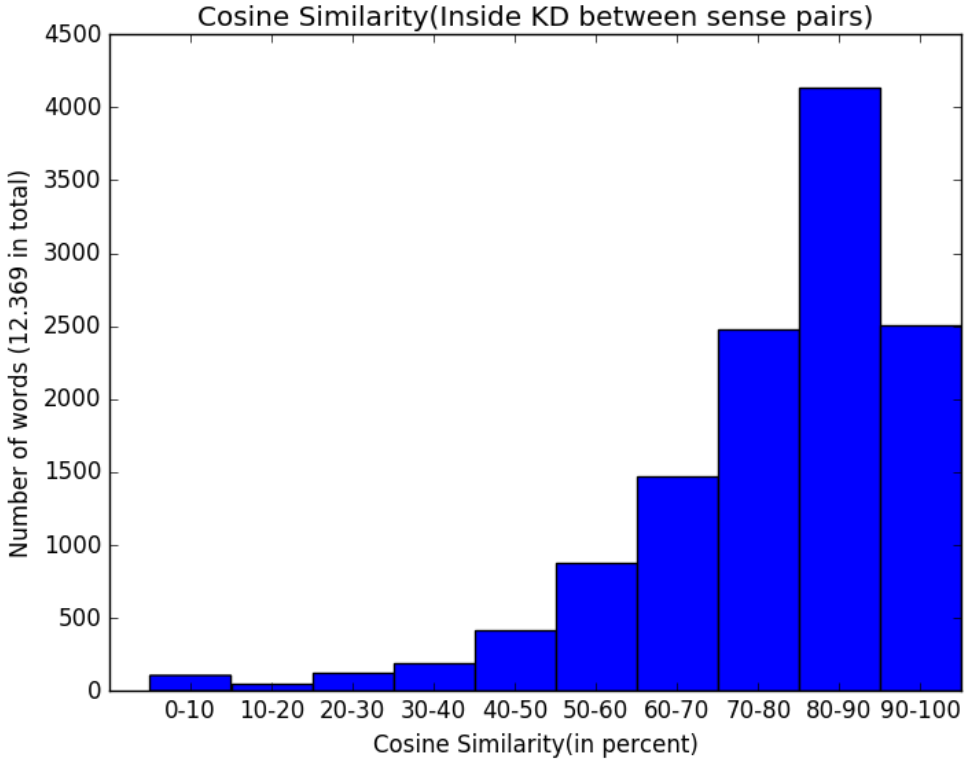


Figure 5.6: Distribution of similarities for word senses inside KD by using Cosine Similarity.

# Chapter 6

# Conclusion

In this project, we designed and implemented our experiments on doing analysis about specified Turkish dictionaries in terms of sense similarities. This similarity analysis will help in future works about "Sense Distinction" decisions while building a new Electronic Turkish dictionary. One of the important concerns about generation of a new electronic dictionary is to eliminate the ambiguous sense definitions. In order to simplify this process, we found orthographic similarities of word senses. Our results shows that there are some extra word sense definitions exist in large amounts for both dictionaries CTD and KD for specific words.

Figure 5.2 indicates that we obtained 16.032 of 27.234 word senses are similar(greater than 0.50) between CTD and KD by using Edit Distance.

Figure 5.5 indicates that we obtained 2.460 of 17.747 word senses are similar(between 0.90-1.00) inside CTD by using Cosine Similarity.

Figure 5.6 indicates that we obtained 2.509 of 12.369 word senses are similar inside KD by using Cosine Similarity.

Figure 5.3 indicates that we obtained 3.819 of 17.747 word senses are similar(greater than 0.50) inside CTD by using Edit Distance.

Figure 5.4 indicates that we obtained 1.836 of 12.369 word senses are similar(greater than 0.50) inside KD by using Edit Distance.

Figure 5.1 indicates that we obtained 14.678 of 27.234 word senses are similar(between 0.90-1.00) between CTD and KD by using Cosine Similarity. These observation proves that word sense definitions cover similar texts between these two dictionaries. In addition, these observations prove that both of two dictionaries have similar texts inside their word senses(between their sense pairs) separately.

## 6.1 Future Works

For the future, the main idea is to focus on generating a sufficient electronic dictionary for Turkish NLP studies. This projects' outputs have useful aspects for future works about making better sense distinction decisions. The results of our analysis indicates the similarity ratios between word senses and it will be highly informative when we will generate a new and optimal 'Electronic Turkish Dictionary. In order to increase the quality of sense distinction decisions, some additional analysis may require to think all aspects of it. For instance, the relations between word senses out of orthography is not covered in this project. We will research on and address some solutions for that topic in future works.

In addition, our TextToVec approach has a trade-off. We used this approach for being able to detect the relational words inside the sense definitions. But in some cases, within the results of this project, the similarity ratio indicates a value more than 0 for sentences have completely different words because they have similar characters. This issue can be handled by obtaining a threshold value. By that reason we will try to determine this optimal threshold value for using the TextToVec approach more efficient for future works related with sense distinction.

# References


[1] A. Kilgarriff, "I dont believe in word senses," *Computers and the Humanities*, vol. 31, no. 2, pp. 91–113, 1997.

[2] M. Palmer, H. T. Dang, and C. Fellbaum, "Making fine-grained and coarse-grained sense distinctions, both manually and automatically," *Natural Language Engineering*, vol. 13, no. 2, pp. 137–163, 2007.

[3] P. Hanks, "The corpus revolution in lexicography," *International Journal of Lexicography*, vol. 25, no. 4, pp. 398–436, 2012.

[4] P. Pantel and D. Lin, "Discovering word senses from text," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 613–619, ACM, 2002.