

**ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES**

PhD THESIS

Erhan TURAN

**AUTOMATIC SYNSET DETECTION FROM TURKISH
DICTIONARY USING CONFIDENCE INDEXING**

DEPARTMENT OF COMPUTER ENGINEERING

ADANA-2020

**ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES**

**AUTOMATIC SYNSET DETECTION FROM TURKISH
DICTIONARY USING CONFIDENCE INDEXING**

Erhan TURAN

Ph.D. THESIS

DEPARTMENT OF COMPUTER ENGINEERING

We certify that the thesis titled above was reviewed and approved for the award of degree of the Doctor of Philosophy by the board of jury on 24/04/2020.

.....
Assoc. Prof. Dr. Umut ORHAN
SUPERVISOR

.....
Prof. Dr. Selma Ayşe ÖZEL
MEMBER

.....
Prof. Dr. Mutlu AVCI
MEMBER

.....
Prof. Dr. Olcay Taner YILDIZ
MEMBER

.....
Asst. Prof. Dr. Ali İNAN
MEMBER

This Ph.D. Thesis is written at the Department of Institute of Natural And Applied Sciences of Çukurova University.

Registration Number:

**Prof. Dr. Mustafa GÖK
Director
Institute of Natural and Applied Sciences**

Not: The usage of the presented specific declarations, tables, figures, and photographs either in this thesis or in any other reference without citation is subject to "The law of Arts and Intellectual Products" number of 5846 of Turkish Republic.

ABSTRACT

PhD THESIS

AUTOMATIC SYNSET DETECTION FROM TURKISH DICTIONARY USING CONFIDENCE INDEXING

Erhan TURAN

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor : Assoc. Prof. Dr. Umut ORHAN
Year: 2020, Pages: 91
Jury : Assoc. Prof. Dr. Umut ORHAN
: Prof. Dr. Selma Ayşe ÖZEL
: Prof. Dr. Mutlu AVCI
: Prof. Dr. Olcay Taner YILDIZ
: Asst. Prof. Dr. Ali İNAN

In this study, a Turkish semantic network is designed from a non-machine-readable monolingual dictionary. Dictionary lemmas and definitions are extracted and processed into a Lemma-Sense weighted bipartite graph model and analyzed for semantic relations. Primary semantic relations of a general semantic network as hypernym, synonym and antonym analyzed based on Lemma-Sense dictionary and added to the semantic network at sense level. Synonym relations are tagged with a confidence level for an improved synset detection. Also, morpho-semantic relations added between the lemmas and their derived and compound lemmas. N-Gram analysis is used to find patterns of any additional semantic relation. These additional semantic relations are supplemented to the semantic network. Finally, synonyms are clustered to form the synsets with a spanning-tree based synset detection algorithm. Synset results are compared with an up-to-date and notable Turkish wordnet.

Key Words: Semantic Network, Wordnet, Turkish, Synset detection, Confidence indexing

ÖZ

DOKTORA TEZİ

**GÜVEN ENDEKSİ KULLANILARAK TÜRKÇE SÖZLÜKTEN EŞ ANLAM
KÜMELERİNİN OTOMATİK TESPİTİ**

Erhan TURAN

**ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

Danışman : Doç. Dr. Umut ORHAN
Yıl: 2020, Sayfa: 91
Jüri : Doç. Dr. Umut ORHAN
: Prof. Dr. Selma Ayşe ÖZEL
: Prof. Dr. Mutlu AVCI
: Prof. Dr. Olcay Taner YILDIZ
: Dr. Öğr. Üyesi Ali İNAN

Bu çalışmada, bir Türkçe anlamsal ağı, bilgisayar okunabilirliği olmayan tek dilli sözlükten tasarlanmıştır. Sözlük madde başları ve tanımları ağırlıklı iki parçalı çizge modeline işlenmiş ve anlamsal ilişkiler açısından analiz edilmiştir. Genel anlamsal ağının üst anlamlı, eş anlamlı ve karşıt anlamlı olarak birincil anlamsal ilişkileri Madde Başı-Anlam sözlüğüne göre analiz edilmiş ve anlam düzeyinde anlamsal ağa eklenmiştir. Eş anlamlı ilişkiler, geliştirilmiş bir eş anlamlılar kümesi tespiti için bir güven seviyesi ile etiketlenir. Ayrıca, madde başları ile bu madde başlarından oluşturulmuş olan türemiş ve bileşik madde başları arasında biçim-anlamsal ilişkiler eklenmiştir. Ayrıca N-Gram analizi, herhangi bir ek anlamsal ilişkinin örüntülerini bulmak için kullanılmış ve örüntüleri bulunan ek anlamsal ilişkiler, anlamsal ağa eklenmiştir. Son olarak, eşanlamlılar, kapsayan ağaç tabanlı eş anlamlılar kümesi algılama algoritması ile eş anlamlılar kümesi oluşturmak için kümelenmiştir. Elde edilen eş anlamlılar kümesi, güncel ve kapsamlı bir Türkçe wordnet ile karşılaştırılmıştır.

Anahtar Kelimeler: Anlamsal Ağ, Wordnet, Türkçe, Eş anlamlılar kümesi algılama, Güven endeksleme

EXPANDED ABSTRACT

Semantics, which is one of the important fields of Natural Language Processing, focus on the semantic analysis of expressions in text with scopes like words, phrases, sentences and documents. And semantic networks are data structures to model lexical semantic units with their relations each other. Dictionaries, especially monolingual ones, which include all concepts of a language, are the most important requirements of semantic studies in natural language processing. Semantic relations in a dictionary can be used, if definitions are designed as machine-readable, automatically in the construction of a semantic network. Preparing a wordnet, a lexical database of semantic relations, for any language involves plenty of time and intense human labor. For most languages other than English, wordnet like networks are generally attempted to be processed automatically with computers.

Semantic networks are prepared using lexical sources as dictionaries, encyclopedias, text corpora or online text mining from the internet. In this study, a Turkish dictionary, Contemporary Turkish Dictionary (CTD), from Turkish Language Association (TLA) is used to create the semantic network for Turkish.

The Lemma-Sense architecture used in this study is inspired by the study of Veronis and Ide (Veronis and Ide 1990). Veronis and Ide (Veronis and Ide 1990) present a semantically more precise network design with a bipartite graph using the lemmas and their senses. Bipartite graph is a graph that nodes of the graph can be divided in two subsets of nodes that all the nodes can be connected to the nodes of the opposite subset of nodes (Buckley and Harary, 1990). Bipartite graph model splits the senses of the lemma and their related lemmas in definitions which is the key concept to manage sense disambiguation. This semantic network model is used in this study for the detection of weighted synonyms pairs. Dictionary lemmas and definitions are transformed into a semantic network through text processing.

Main semantic relations of a general semantic network, hypernym, synonym and antonym, are analyzed based on Lemma-Sense dictionary and added to the semantic network at sense level. Also, morpho-semantic relations are added between the lemmas and their derived and compound lemmas.

Definitions in the dictionary may have distinguishable patterns for some words or word groups which can be useful to analyze relationships between words and to find forms of derivative suffix with definition patterns. In this study, an application called N-Gram Analyzer is designed to analyze explanations for word-based n-grams. After applying n-gram extraction, n-grams are found from 1-gram to the longest n-gram, 56-gram. And N-Gram Analyzer calculates the Maximum Likelihood Estimation for the n-gram to find the longest n-gram pattern from subsets of a proper n-gram. N-gram analysis is used to find patterns of any additional semantic relation. These additional semantic relations are supplemented to the semantic network.

In this study, antonym relations are analyzed with sense to sense level with two different methods. In first method when a definition of lemma, S_1 , contains the antonym pattern, the antonym is extracted and searched for its definitions to find the correct antonym sense, S_2 . If one of the senses of antonym references back to the lemma with sense S_1 , then S_1 and S_2 are antonyms. The second method used to find antonyms is based on the Presence and Absence relations explained in previous sections. These relations are the adjective forms of a noun representing the state of existence of the noun. Presence relations has a pattern "[noun] olan" and Absence relations has a pattern "[noun] olmayan" and these patterns are compared if they have identical nouns.

Synonym relations are tagged with a confidence level for an improved synset detection. Some evaluations should be made on the bipartite graph obtained by passing the definition sentences through text processing. For example, suppose that the lemma Y is a synonym in the description of sense X_i of lemma X . In the semantic network, it is a serious problem to determine which of the senses of

Lemma Y link a synonym relation from the sense X_i . To solve this problem, it is checked whether there is a direct reference from each Sense Y_j to the sense X_i , in other words, whether it is regular. Each relation gets a confidence index value based on the structure of its definition.

Finally, synonyms are clustered to form the synsets with a spanning-tree based synset detection algorithm. After analyzing the semantic network, labeling synonym relations with confidence levels and choosing some synonym relations depending on their confidence level, converting the chosen directed synonym relations into undirected ones, and then making spanning tree-based synset detection on the undirected graph is proposed for the first time in this study. Synset results are compared with an up-to-date and notable Turkish wordnet, KeNet(Ehsani, Solak, and Yildiz 2018) .



GENİŞLETİLMİŞ ÖZET

Doğal Dil İşlemenin önemli alanlarından biri olan anlambilim, metindeki ifadelerin kelimeler, deyimler, cümleler veya belgeler kapsamında anlamsal analizine odaklanır. Anlamsal Ağlar, sözcük birimlerinin birbirleriyle olan ilişkilerini modellemek için kullanılan veri yapılarıdır. Sözlükler, özellikle bir dilin tüm kavramlarını içeren tek dilli olanlar, doğal dil işlemede anlamsal çalışmaların en önemli gereksinimleridir. Bir sözlükteki anlamsal ilişkiler, tanımlar makine tarafından okunabilir olarak tasarlandysa, otomatik olarak bir anlamsal ağı oluşturulmasında kullanılabilir.

Anlamsal ilişkilerin sözcüksel bir veritabanı olan wordneti herhangi bir dil için hazırlamak çok zaman ve yoğun insan emeği gerektirir. İngilizce dışındaki birçok dilde, wordnet benzeri ağlar genellikle bilgisayarlarla otomatik olarak yapılmaya çalışılır.

Anlamsal ağlar, sözlükler, ansiklopediler, metin derlemleri veya internetten çevrimiçi metin madenciliği gibi kaynakların biri veya bir kaçı kullanılarak hazırlanır. Bu çalışmada Türkçe için anlambilimsel ağ oluşturmak amacıyla bir tek dilli Türkçe sözlük olan Türk Dil Kurumuna ait, Güncel Türkçe Sözlük kullanılmıştır.

Bu çalışmada kullanılan Madde Başlı-Anlam mimarisi, Veronis ve Ide'nin çalışmasından esinlenmiştir (Veronis ve Ide 1990). Veronis ve Ide (Veronis ve Ide 1990), madde başlarını ve onlara ait olan anlamları kullanarak iki parçalı bir çizge tasarımına sahip anlamsal olarak daha hassas bir ağ tasarımı sunar. İki parçalı çizge, çizgenin düğümlerinin, tüm düğümlerin karşıt düğüm alt kümesinin düğümlerine bağlanabileceği iki farklı düğüm alt kümesine ayrılabilirdiği bir çizgedir (Buckey ve Harary, 1990). İki parçalı çizge tasarımı, bir madde başının diğer madde başları ile olan ilişkisel bağlantılarını madde başının sahip olduğu anlamlara ayırıştırılmasını sağlayarak anlam karmaşasına karşı önemli bir üstünlük sağlar. Bu anlamsal ağ tasarımı, bu çalışmada ağırlıklı eş anlamlı çiftlerinin tespiti

iin kullanılmıřtır. Sözlükteki madde bařları ve anlamları metin iřleme yoluyla anlamsal bir aęa dönüřtürölür.

Genel bir anlamsal aęının ana anlamsal iliřkileri, üř anlamlı, eř anlamlı ve karřıt anlamlı, Madde bařı-Anlam sözlüęü temel alınarak analiz edilir ve anlam düzeyinde anlamsal aęa eklenir. Ayrıca, madde bařları ile bunların türetilmiř ve bileřik madde bařları arasında biçim-anlamsal iliřkiler eklenir.

Sözlükteki tanımlar, sözcükler arasındaki anlamsal iliřkiler iin bazı sözcükler veya sözcük grupları ile tanım örüntülerine sahip olabilir ve bu ayırt edici örüntüler birok anlamsal iliřki ortaya ıkarırken türemiř ve karřıt anlam iliřkilerinde yardımcı iliřkiler saęlayabilirler. Bu alıřmada, tanımlardan elde edilen kelime tabanlı n-gramları analiz etmek iin N-Gram Analyzer adı verilen bir uygulama tasarlanmıřtır. N-gramlar uygulama ile elde edildikten sonra, 1-gramdan en uzun n-gram 56-gram'a kadar n-gramlar bulunmuřtur. Ve N-Gram Anaylzer, uygun bir n-gramın alt kümelerinden en uzun n-gram örüntüsünü bulmak iin n-gramlar üzerinde Maksimum Olabilirlik Tahminini hesaplanmıřtır. N-gram analizi, herhangi bir ek anlamsal iliřkinin örüntülerini bulmak iin de kullanılmıř ve bulunan anlamsal iliřkiler, anlamsal aęa eklenmiřtir.

Bu alıřmada, karřıt anlam iliřkileri iki farklı yöntemle anlam düzeyinde incelenmiřtir. İlk yöntemde, bir madde bařının anlamı, A_1 , karřıt anlam örüntüsü ierdięinde, karřıt anlam ıkarılır ve doęru karřıt madde bařının anlamı A_2 'yi bulmak iin karřıt anlamlı madde bařının tanımları arařtırılır. Karřıt anlamlı madde bařının tanımlarından biri karřıt anlam örüntüsü ile A_1 'i iřaret ediyorsa A_1 anlamı ile A_2 anlamları karřıt anlamlı olarak iliřkilendirilir. Karřıt anlamları bulmak iin kullanılan ikinci yöntem ise, Varlık (Presence) ve Yokluk (Absence) iliřkilerine dayanmaktadır. Bu iliřkiler, herhangi bir isim madde bařının var olma durumlarını iřaret eden sıfat madde bařlarına yapılan baęlantılardır. Varlık iliřkileri "*[isim] olan*" kalıbına sahiptir ve Yokluk iliřkileri "*[isim] olmayan*" kalıbına sahiptir ve bu iki kalıpta geen isim aynı ise bu kalıplara sahip olan anlamlar karřıt anlamlı olarak iliřkilendirilir.

Eş anlamlı ilişkiler, gelişmiş bir eş anlamlılar kümesi algılaması için bir güven düzeyi ile etiketlenir. Tanım cümlelerinin metin işlemeyle elde edilen iki parçalı çizge üzerinde bazı değerlendirmeler yapılmalıdır. Örneğin, madde başı Y 'nin, madde başı X 'in X_i anlamının açıklamasında bir eşanlamlı olduğunu belirlenmiş ise, anlamsal ağda, madde başı Y anlamlarından hangisinin X_i anlamı ile eş anlamlı bir ilişki kurduğunu belirlemek ciddi bir sorundur. Bu sorunu çözmek için, her bir anlam Y_j 'den X_i anlamına doğrudan bir referans olup olmadığı, diğer bir deyişle düzenli bir eş anlamlı çift olup olmadığı kontrol edilir. Her eş anlam ilişkisi, anlama ait tanımın yapısına bağlı olarak bir güven endeksi değeri alır.

Son olarak, eşanlamlılar, kapsayan ağaç tabanlı eş anlamlılar kümesi algılama algoritması ile eş anlamlılar kümesi oluşturmak için kümelenir. İlk kez bu çalışmada özgün olarak, anlamsal ağı analiz ettikten sonra, eşanlamlı ilişkilerini güven düzeyleri ile etiketlenmesi, güven düzeylerine bağlı olarak bazı eşanlamlı ilişkilerin seçilmesi, seçilen yönlü eşanlamlı ilişkilerin yönsüz eş anlam ilişkilere dönüştürülmesi ve ardından yönsüz çizgede kapsayan ağaç tabanlı eş anlamlılar kümesi tespiti yapılması önerilmiştir. Eş anlamlılar kümelerinin sonuçları, güncel ve yetkin bir Türkçe wordnet olan KeNet (Ehsani, Solak ve Yıldız 2018) ile karşılaştırılmıştır.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Assoc. Prof. Dr. Umut ORHAN for his supervision, encouragement, patience, motivations and useful suggestions for the successful completion of this work.

I would like to thank members of the Ph.D. thesis jury, Prof. Dr. Selma Ayşe ÖZEL, Prof. Dr. Mutlu AVCI, Prof. Dr. Olcay Taner YILDIZ, and Assist. Prof. Dr. Ali İNAN for their valuable help, support and suggestions.

I would like to express my sincere appreciation to TUBİTAK for their financial support with the project number 215E256 in the process of my Ph.D. education.

I am also thankful to Dr. Çağatay Neftali TÜLÜ and Dr. Enis ARSLAN for their support and motivation.

Finally, I would like to thank my family who have supported me throughout the entire thesis period, both by their love and encouragement.

CONTENTS	PAGE
ABSTRACT.....	I
ÖZ.....	II
EXPANDED ABSTRACT	III
GENİŞLETİLMİŞ ÖZET	VII
ACKNOWLEDGMENTS	XI
CONTENTS.....	XII
LIST OF TABLES	XIV
LIST OF FIGURES	XVI
LIST OF ABBREVIATIONS	XVIII
1. INTRODUCTION	1
1.1. The Aims and Objectives of This Thesis	4
1.2. Our Contribution	5
1.3. Thesis Organization	5
2. RELATED WORKS	7
3. MATERIALS AND METHODS.....	15
3.1. Dictionary Data And Preprocessing.....	16
3.1.1. N-Gram Analysis	31
3.1.2. Analysis of Derived and Antonym Words.....	39
3.2. Creating the Lemma-Sense Network	43
3.3. Labeling of Synonym Pairs with Confidence Indexing	49
3.4. Evaluation of Confidence Indexed Pairs.....	56
3.5. Spanning Tree-based Synset Detection.....	61
4. RESULTS AND DISCUSSIONS.....	69
4.1. Semantic Network Analysis.....	69
4.2. Indexing Analysis of Synonym Relations.....	73
4.3. Comparison of the Results	78

5. CONCLUSION.....	81
REFERENCES	83
CURRICULUM VITAE.....	91



LIST OF TABLES	PAGE
Table 3.1 Statistics about Polysemous and Univocal Lemmas.....	18
Table 3.2 Sense numbers of Single-word Lemmas and MWEs.....	19
Table 3.3 Property list of a lemma	20
Table 3.4. Enumerated values of Expression property of the lemma	22
Table 3.5. Irregular suffix patterns for vowel drop of Inflection property	23
Table 3.6. Irregular Suffix Patterns for loan words with twin consonants	23
Table 3.7. Irregular Suffix Patterns for Suffix with Connecting Sounds.....	24
Table 3.8. Irregular Suffix Patterns for Suffix with Consonant Assimilation	24
Table 3.9. Loan words numbers with their origin language	25
Table 3.10. Type property values with their frequency in the dataset	26
Table 3.11. Usage property values with their numbers in the dataset.....	27
Table 3.12. Disciplines class types of Term property	28
Table 3.13. Objective property tags	29
Table 3.14. N-Gram Distribution for the definitions from CTD dataset	33
Table 3.15. Patterns from CTD found by n-gram analysis for Group Of.....	36
Table 3.16. Patterns from CTD found by n-gram analysis for Hypernymy ...	37
Table 3.17. Patterns from CTD found by n-gram analysis	38
Table 3.18. A synonym pair example for 0-2	56
Table 3.19. A synonym pair example for 0-3	57
Table 3.20. A synonym pair example for 0-4	57
Table 3.21. A synonym pair example for 0-5	58
Table 3.22. A synonym pair example for 1-5	59
Table 3.23. A synonym pair example for 2-5	60

Table 3.24. A synonym pair example for 5-5	61
Table 3.25. Seven lemmas and their definitions for synset detection.....	66
Table 4.1. Relations linked between nouns senses	70
Table 4.2. Adjective and Adverb based semantic relations	71
Table 4.3. Verb based semantic relations.....	71
Table 4.4. The distribution of found synonym relations.....	73
Table 4.5 An example for POS tag mismatch.....	74
Table 4.6. The lemmas and the synonym definitions of the second example.	75
Table 4.7. The numbers of correct matched senses by different distance factors	77
Table 4.8. The lemmas and the definitions of the third example.....	77
Table 4.9. Comparison of two studies for different confidence levels	79

LIST OF FIGURES	PAGE
Figure 1.1. Semantic relations from Princeton WordNet.....	1
Figure 2.1. Bipolar adjective structure (Fellbaum, 1998)	7
Figure 2.2. A subgraph form Stanchev's phrase graph (Stanchev 2012)	10
Figure 3.1. Summary of the study.....	15
Figure 3.2. Output of lemma "öğrenci" (student) from Online CTD	17
Figure 3.3. Record of lemma "öğrenci" (student) in NoSQL database	17
Figure 3.4. NoSQL data structure of Lemmas "ev" and "kurt"	21
Figure 3.5. Output of lemma "alet" (tool) from online CTD	27
Figure 3.6. Document structure of n-gram data in NoSQL database	32
Figure 3.7. Graphical user interface of NGram Analyzer	35
Figure 3.8. Bipartite graph model for semantic network	45
Figure 3.9. A subgraph with three lemmas based on Lemma-Sense architecture ..	46
Figure 3.10. A synonym sense ambiguity of multiple ci5 relations.....	53
Figure 3.11. Two synsets after disambiguation of ci5 relations	54
Figure 3.12. A three-word synset.....	62
Figure 3.13. Regular synonym relations among the seven senses	67
Figure 4.1. Root lemma "göz" and some of its derivative lemmas	72
Figure 4.2. The undirected graph of first example.....	74
Figure 4.3. The undirected graph of second example	76
Figure 4.4. The undirected graph of the third example.....	78



LIST OF ABBREVIATIONS

3SG	: Third Singular Person
API	: Application Programming Interface
BFS	: Breadth First Search
CDV	: Controlled Defining Vocabulary
CTD	: Contemporary Turkish Dictionary
DFS	: Depth-First Search
GUI	: Graphical User Interface
MLE	: Maximum Likelihood Estimation
MWE	: Multiword Expression
NLP	: Natural Language Processing
POS	: Part of Speech
PWN	: Princeton WordNet
SCC	: Strongly Connected Components
SOV	: Subject Object Verb
SPF	: Shortest Path First
TLA	: Turkish Language Association
WSD	: Word Sense Disambiguation
XML	: Extensible Markup Language



1. INTRODUCTION

Dictionaries, especially monolingual ones, which include all concepts of a language, are the most important requirements of semantic studies in natural language processing. However, dictionaries are required to be converted into semantic networks for computer-based studies. Semantic relations in a dictionary can be used, if definitions are designed as machine-readable, automatically in the construction of a semantic network. Otherwise, preparing a semantic network by experts is a highly labor-intensive process. Primary semantic relations in Princeton WordNet (PWN), first semantic network for English, are shown in the Figure 1.1.

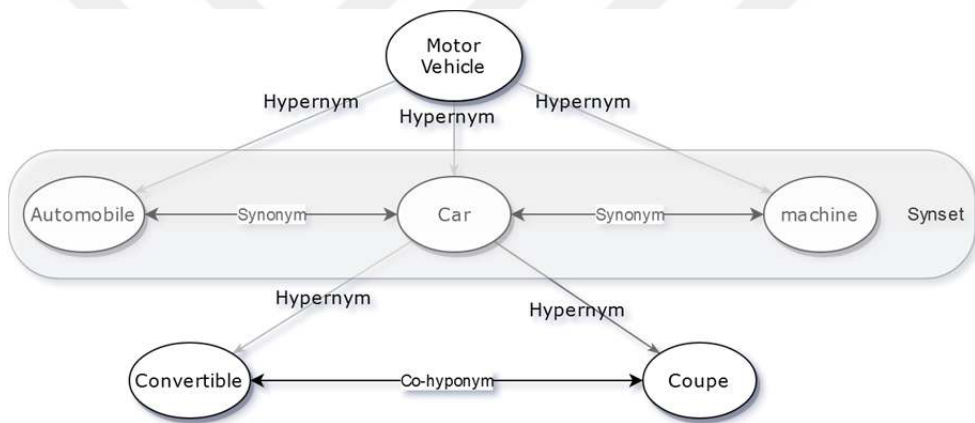


Figure 1.1. Semantic relations from Princeton WordNet

In Figure 1.1, nodes are the concepts of a natural language and the links are the semantic relations between these concepts. Synonyms in Figure 1.1, Automobile, Car and Machine form a synonym set or *synset* in short. Synsets in a semantic network such example in Figure 1.1, construct the base structure of a semantic network. Hypernym relations form the taxonomy of the concepts in a semantic network.

Semantic networks have been raised as a new field in computer science with studies that model the human mind in the late 1960s. The study of Katz and

Fodor (Katz and Fodor 1963) on the structure of semantic theory reveals the basic concepts of the semantic network. Quillian (Quillian 1969), on the other hand, designed a computer program with LISP for semantic analysis. Woods and Beranek (Woods and Beranek 1975), worked on semantic networks for notation of the information people have. In their work, Collins and Loftus (Collins and Loftus 1975) focused on the distance between concepts in semantic networks. In the early 2000s, Widdows et al. (Widdows, Cederberg, and Dorow 2002; Widdows and Dorow 2002; 2005) provided examples of graph models for the semantic network. Beside all these studies, Princeton WordNet (PWN) designed for the English language is the first official semantic network and has been accepted as a pioneering study in the literature (Fellbaum 1998).

Preparing wordnet for any language involves plenty of time and intense human labor. For most languages other than English, wordnet-like networks are generally attempted to be done automatically with computers. In computer-based wordnet studies, two different approaches are generally used. The first method is the extraction of basic semantic relationships with text processing methods directly from a monolingual dictionary and/or encyclopedia data of that language (Oliveira, Santos, and Gomes 2009; Gonalo Oliveira and Gomes 2014; Alexeyevsky and Temchenko 2016; Ehsani, Solak, and Yildiz 2018). The other method is based on the translation of PWN using a multilingual dictionary (Vossen 1998; Sofia et al. 2002; Bilgin, Cetinoglu, and Oflazer 2004; Sagot and Fiřer 2008; Putra, Arfan, and Manurung 2008; Thoongsup et al. 2009; Oliver and Climent 2012; Bond and Foster 2013; Ercan and Haziyeve 2019). In the first approach, the goal is to generate wordnet using monolingual resources of the target language and then synchronize it with PWN as much as possible. In the second approach, wordnet for the language is produced by translating it directly from PWN using a multilingual dictionary. But the success of the translation process depends on the semantic matching between the languages. On the other hand, the struggle to have global validity of the wordnet produced for any language sometimes makes it difficult to preserve

language-specific concepts such as connotation, figurative meanings and idioms (Kashgary 2011). In such cases, if necessary, manual matching methods used to adjust the concepts (Bosch and Griesel 2017). Therefore, in order to prepare comprehensive wordnet for a language, it is almost compulsory to use monolingual materials (dictionary and/or encyclopedia) that fully cover the concepts of that language (Gonalo Oliveira and Gomes 2014; Alexeyevsky and Temchenko 2016; Ehsani, Solak, and Yildiz 2018). In this context, GermaNet (Hamp and Feldweg 1997) launched for German is a good example. It is started entirely based on the manual inputs of experts, as in the PWN project, and then included in the EuroWordNet project (Vossen 1998). GermaNet contains all the words and relationships specific to German thanks to its manual creation, while it is aligned with PWN owing to the EuroWordNet project.

In the literature, there are many wordnet projects where both methods mentioned above are used in the same project. The wordnets obtained in these studies seem more inclusive in terms of the number of synsets they contain, as they start from a monolingual resource. For example, BalkaNet (Bilgin, Cetinoglu, and Oflazer 2004), which is started to prepare Turkish wordnet aligned with PWN, contains many fewer synsets compared to KeNet (Ehsani, Solak, and Yildiz 2018), which is recently prepared with monolingual dictionary data. Besides, the Onto.PT (Gonalo Oliveira and Gomes 2014) study prepared for Portuguese has three times more synsets even than the largest of other semantic networks in Portuguese, as well as different relationships not found in PWN (de Paiva and Real 2016).

On the other hand, various applications have been developed to manually intervene to translations in studies aiming to prepare wordnet aligned to PWN (Horak et al. 2006; Finlayson 2014). However, as the target language's conceptual difference from the English language increased, the process became more difficult. Also, the translation approach could not be applied for Multi-Word Expressions (MWE) specific to the relevant language that are not included in the PWN as a concept, and these terms, which cannot be translated, required adding new

comments (Kashgary 2011; Bosch and Griesel 2017). The addition of language-specific MWEs and relationships to wordnet has been one of the main advantages of studies using monolingual resources. However, using non-machine-readable monolingual resources can make it difficult to extract relationships with parsing (Alexeyevsky and Temchenko 2016; Ehsani, Solak, and Yildiz 2018). Therefore, determining how readable a monolingual resource is by machines should be the first step to be considered in wordnet design.

1.1. The Aims and Objectives of This Thesis

Turkish wordnet studies are begun with Bilgin et al. (Bilgin, Cetinoglu, and Oflazer 2004) and proceed with Amasyalı (Amasyalı 2005), however, stalled for long period until Ehsani et al. (Ehsani, Solak, and Yildiz 2018) study, KeNet. Nowadays, only KeNet XML data is publicly available to use as a Turkish wordnet. In this study, it is aimed to design a semantic network based on but not limited to the main relations of a wordnet. It is a very labor-intensive task to create a semantic network with manually. And for Turkish with over 90,000 lemmas and 120,000 senses, a machine-based approach is considered to design the semantic network. In this study, unlike PWN data model, semantic network is created on a graph model with all POS type lemmas in a dictionary and unlike KeNet, semantic network is dealt with MWEs and derived lemmas, to be suited for Turkish.

Semantic relations are extracted from a lexicon for Turkish since early 2000s (Amasyalı 2005; Güngör and Güngör 2007; Şerbetçi, Orhan, and Pehlivan 2011; Yazıcı and Amasyalı 2011). Synonym relations are considered in the first place to detect between lemmas. In this study, Hypernym and Group of relations and other relations extracted by a comprehensive n-gram analysis on Contemporary Turkish Dictionary to find every appropriate semantic relations for Turkish. Compound and Derived relations between lemmas are extracted to supplement the

network with morpho-semantic relations. Antonym and synonym relations are created between senses to consider word sense disambiguation.

1.2. Our Contribution

Studies on wordnet designing and semantic relation extracting for Turkish is begun from early 2000s and universal methodologies for a natural language are applied in those studies. And those methodologies are also adopted in this study. However, creating a semantic network directly on a graph model with all lemmas including MWEs from a dictionary; extracting semantic relations based on n-gram analysis; proposing an easy to compute semantic distance algorithm; adding derive, compound and phrase relations for morpho-semantic relations for Turkish are subsidiary contributions of this study.

Using a confidence index for synonym relations and analyzing synonyms according to these index values to detect synsets automatically and revealing semantic errors in these synsets is the primary contribution of this study.

1.3. Thesis Organization

This thesis is organized as follows:

In Section 2, a literature overview of studies on semantic networks and wordnets is provided. Studies on the extraction of semantic relations from the monolingual dictionary are outlined and these methods applied on many languages in the world, especially Turkish, are explained.

In Section 3, methods and materials of this study explained in details. Firstly, a monolingual dictionary is analyzed and preprocessed for creating a semantic network. Dictionary data structure described with properties of lemmas and their definitions. Dictionary data is used to obtain patterns for semantic relations with N-Gram analysis. Hypernym, Group Of and additional semantic relations extracted with these patterns. Semantic network based on Lemma-Sense

bipartite graph design is presented and linking morpho-semantic relations between lemmas and their compound and derived word lemmas is explained. Then, appending Hypernym, Group of, Antonym and additional semantic relations on the graph model explained in details. Finally, labeling synonyms with a confidence index then detecting the synsets from these labeled synonyms are explained with our confidence indexing method and spanning-tree based synset detection algorithm.

In Section 4, results and statistics about semantic relations are presented and semantic errors revealed by synset detection algorithm and results of synset analysis are explained in details with example cases.

Finally, in the Conclusion section, proposed methods explained in previous sections are discussed for their contributions and inadequacies. And for future studies, some suggestions are presented by interpreting the problems experienced.

2. RELATED WORKS

Semantics, which is one of the important fields of Natural Language Processing, focus on the semantic analysis of expressions in text with scopes like words, phrases, sentences and documents. *Lexical Semantics* studies focusing on the analysis of words, affixes and compound words research for essential methods and data models for a similar semantic model of human thinking in computer systems. And *Semantic Networks* are data structures to model lexical semantic units with their relations each other. Although not designed as a graph model, Princeton WordNet (PWN) (Fellbaum 1998) created with human labor for the English language, is a pioneering semantic network frequently used in many semantic studies as an important tool. PWN has primary semantic relations for different part of speech (POS) words. In Figure 2.1, *fast* and *slow* are antonyms in the PWN and *prompt* is similar to *fast*. There is no sense which directly reference *prompt* as an antonym in PWN. However, a suitable antonym can be found by using similarity and antonym. In PWN, *prompt* is similar to *fast* which is direct antonym of *slow* and any similar sense to *slow* can be an antonym for *prompt*.

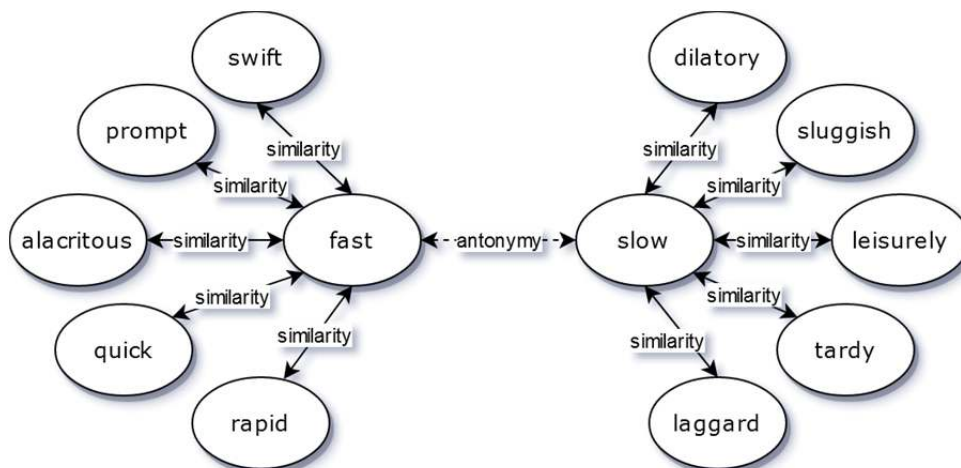


Figure 2.1. Bipolar adjective structure (Fellbaum, 1998)

The first semantic network study for the Turkish natural language is the BalkaNet(Sofia et al. 2002) subproject study initiated by Bilgin and colleagues (Bilgin, Çetinoğlu, and Oflazer 2004) inspired by English WordNet. Turkish WordNet in BalkaNet is a network that needs to be improved due to a very limited number of words coverage, although promising, for the Turkish language.

On the other hand, Amasyalı (Amasyalı 2005), designed a Turkish semantic network automatically in his work which contains various methods, however has not achieved an effective result due to poor translation between English and Turkish. Although many researchers have been working on it, it can be said that an effective semantic network such as WordNet still in progress for Turkish.

A semantic network created by experts requires investment in a long and tedious project, although it is highly reliable. In this stage, it may be possible to design a semantic network in a short time and automatically using the methods that computer science can offer. Dictionaries and encyclopedias can be used as input data during the design of such a semantic network and can be supplemented with corpus and other documents to be collected from the internet.

Dictionaries and encyclopedias have a semantic relationship between articles and their definition sentences. In a dictionary, an article may refer to synonyms, hypernyms, antonyms and other semantically related words in its definitions. Although these relations protect their existence in encyclopedias, their direct use can sometimes be troublesome.

Designing a semantic network by taking advantage of the dictionaries goes back to the study of Chodorow and his colleagues (Chodorow, Byrd, and Heidorn 1985) on the English language. Work for Turkish natural language has begun later, and most comprehensive studies have recently occurred (Güngör and Güngör 2007; Orhan et al. 2011; Şerbetçi, Orhan, and Pehlivan 2011; Yazıcı and Amasyalı 2011).

Veronis and Ide (Veronis and Ide 1990) presented a network model based on "*lemma - sense*" connections with Collins English Dictionary definitions in their

work. Each lemma is added as a lemma node to the network, and linked to its definitions with sense nodes, while the sense nodes are linked to the nodes of the other lemmas mentioned in the definition. These links between different type of nodes are the Excitatory Links which connects Lemma nodes together over the semantically relevant sense nodes. Inhibitory Links are the connection between the senses of the same lemma to send inhibition each other to compete in word sense disambiguation. In Figure 2.2, a part of the network of Veronis and Ide's study with lemma and sense nodes and links between each other. Disambiguating between senses of a lemma is processed by using links between any two lemma to find the closest path with spreading activation model over the network.

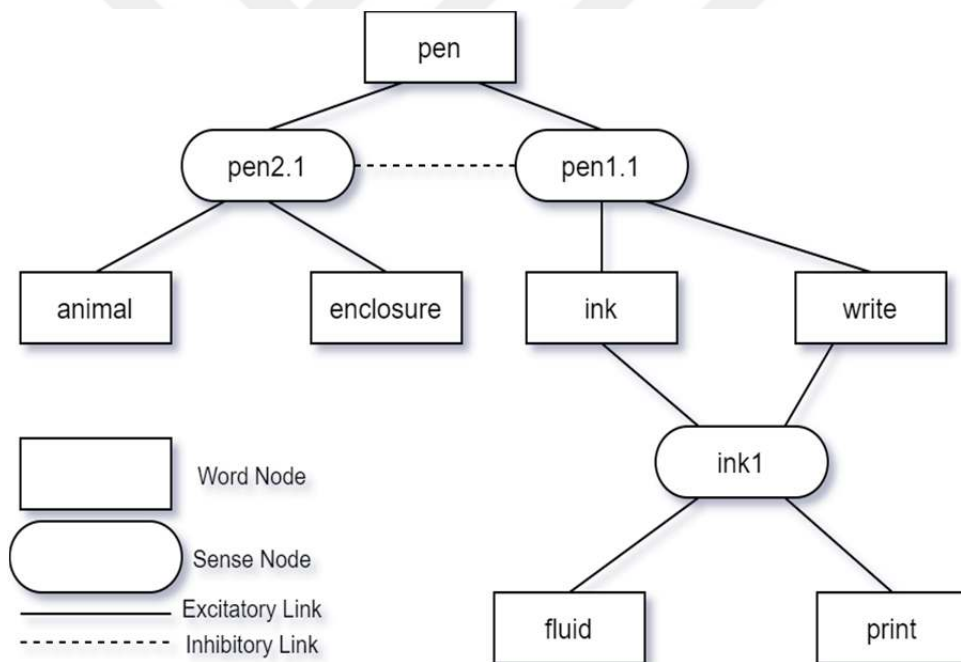


Figure 2.2. Network model for lemmas and their definitions (Veronis and Ide, 1990)

In another study, Widdow and Dorow (Widdows and Dorow 2002) designed a semantic network using a tagged corpus and compared it with WordNet.

In a recent study, Stanchev (Stanchev 2012) designed a semantic network by linking each lemma in the WordNet dataset with the definitions attached to it in a graph model, then linking the words in the example sentences of definitions.

In Figure 2.3, a subgraph from Stanchev's phrase graph containing lemma and sense nodes with weighted edges. PWN has a frequency for senses of each lemma which is used as a probability weight between a lemma and its senses in phrase graph. Furthermore, each sense has links to the lemmas in its definition with a weight. And these links have a weight calculated by frequency of the non-noise lemma appears in the sense's definition divided by the total appearance number of non-noise lemmas in the sense's definition multiplied by a_1 , another parameter which is given in the study. The values of a_1 and a_3 parameters are given by Stanchev.

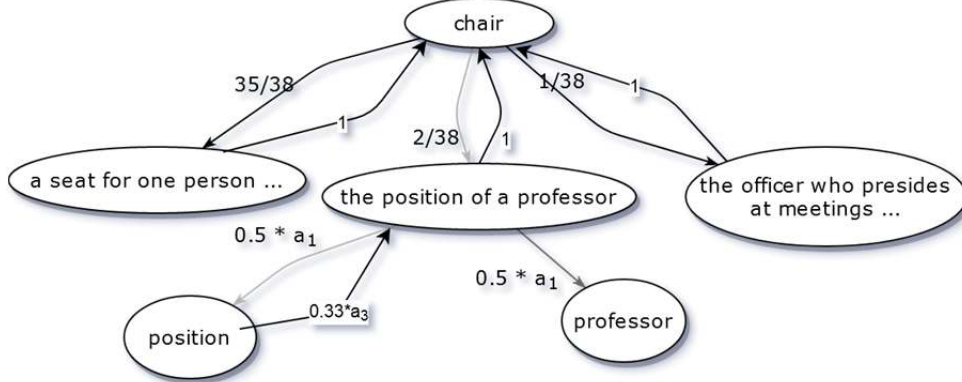


Figure 2.3. A subgraph form Stanchev's phrase graph (Stanchev 2012)

The main idea in phrase graph structure is to calculate the weights of the edges by using WordNet's frequency data on lemmas and definitions. These weights are used to compute the semantic distance between the lemma nodes.

Budanitsky and Hirst (Budanitsky and Hirst 2006) in a comprehensive study, compared the methods of measuring similarity between nodes on semantic networks and mentioned studies of Kozima and Furugori (Kozima and Furugori 1993) and Kozima and Ito (Kozima and Ito 1997) as examples of dictionary-based approaches.

In their work, Kozima and Furugori designed a semantic network with 2,861 English words taken from the English dictionary and calculated similarity on this network by using the method of "*spread activation*". In the other study, Kozima and Ito produced P-vectors by spreading activation on the semantic network that constructed from subgroup of English words and then formed semantic space from the Q-vectors obtained by principal component analysis of P-vectors.

Both studies emphasize the selection of words with intensive semantic bonds within the English dictionary and mention that the similarity calculations in the methods can be poorly effected by sparsity for a semantic network that created with all of the words in a dictionary. Another measure of similarity is proposed by Thorat and Choudhari (Thorat and Choudhari 2016) for inverse dictionary design by the similarity distance method. In this study, the frequency of a lemma appearing in the definitions of the dictionary effects the similarity distance, for a lemma, high frequency values will increase its similarity distance to other words to neutralize the lemma node in case of being function word.

On studies in the domain of Turkish natural language, dictionary-based methods are frequently used to create semantic networks, on the other hand, there is still no reliable semantic network for general usage in the literature. Güngör and Güngör (Güngör and Güngör 2007), Orhan and her colleagues (Orhan et al. 2011), Şerbetçi and her colleagues (Şerbetçi, Orhan, and Pehlivan 2011) and Yazıcı and Amasyalı (Yazıcı and Amasyalı 2011) studied on extracting semantic relations from Turkish dictionary using rule based text processing.

Güngör and Güngör is proposed a heuristic algorithm to extract hypernym relations from the definitions of a dictionary. They used a general pattern in the definitions such as $(w^* \text{ hype}) (, w^* \text{ hype})^* (, \text{ syn})^*$ in regular grammar. In the pattern w denotes any word in the definition, while hype and syn are the words correspond to a possible hypernym and the synonym of the word. A definition usually ends with a synonym if exist, when possible synonyms are trimmed, the last word of a pattern is a candidate for hypernym. Güngör and Güngör determined several rules after analyzing the dictionary for noun lemmas. They found 11 rules categorized in three groups. First group is the rules that determines hypernym according to the noun's surface form. Second group is the rules determines hypernym according to the category of the noun defined in the dictionary. Third group is the rules that determines hypernym according to the definition of the noun in the dictionary. In this study, after n-gram analysis, patterns for hypernym relations are found according to the definitions similar to Güngör and Güngör's third group for the hypernym rules. Güngör and Güngör used these hypernyms to construct a hierarchical structure of nouns in the dictionary. And their hierarchical structure was formed with 72 levels, which is far more levels than expected hierarchy for nouns in a natural language. This situation is caused by improper hypernym model in the definitions and lack of a word sense disambiguation module. Improper hypernym model usually is encountered by defining a word with a higher level hypernym than the lowest hypernym of that word. For example, lemma “*kedi*” (*cat*) has a definition such as “*Kedigillerden, memeli, köpek dişleri iyi gelişmiş, çevik ve kuvvetli, evcil, küçük hayvan, pisik*” in Contemporary Turkish Dictionary (CTD). And there are two hypernym candidate in the definition, first one is the lowest level hypernym lemma “*kedigiller*” (*felines*) and the second one is a higher level hypernym lemma “*hayvan*” (*animal*) for lemma “*cat*”. Synonym extracting is applied with analyzing the same general pattern. However, lack of a proper word sense disambiguation module, found synonym relations must be validated manually.

Orhan and her colleagues (Orhan et al. 2011) and Şerbetçi and her colleagues (Şerbetçi, Orhan, and Pehlivan 2011) are proposed semantic relation extracting from a Turkish dictionary based on pattern rules for dictionary definitions. Both studies used similar patterns as in Güngör and Güngör study although with more semantic relations such as *Kind-Of*, *Amount-Of*, *Group-Of*, *Member-Of*, *Is-a* and *Has-a*. In Şerbetçi and her colleagues' study, using morpho-semantic patterns to take advantage of morphological structure of Turkish language is also advised.

Yazıcı and Amasyalı (Yazıcı and Amasyalı 2011) extracted semantic relations from Turkish Dictionary with predefined text processing rules. They obtained synonymy relations with two different pattern approaches. In the first pattern approach, they extracted synonyms from end of the definition separated with commas and created synonymy relations between the word of the definition and the synonyms at the end the definition. In the second pattern approach they used the same pattern separated with commas in the other parts of the definition and created synonymy relations between the words extracted from the pattern while omitting the word of the definition. But they did not resolve the flaw in the second approach the probability of tagging co-hyponyms as synonyms.

The overall drawback of all these studies, which focus on an automatic semantic network design using dictionary data, is that no reliable automatic method used to validate the results. Thus, lack of a validation method compromises the reliability of the semantic relations extracted in these studies. In our study, CTD is morpho-semantically analyzed to extract compound, MWEs and derived lemmas in the dictionary data. Synonym and antonym relations are extracted by both way references to validate the semantic relations.

A general n-gram analysis applied on dictionary data to find all possible patterns for any kind of semantic relation in the dictionary. Synonym relations are extracted with considering word sense disambiguation using Mention distance over

Lemma-Sense network. Synsets are detected with indexed synonyms to find reliable synsets and semantic problems.



3. MATERIALS AND METHODS

In the study, Turkish dictionary definitions turn into semantic relations through three stages (text processing, graph generating, and semantic relation analysis). The study is summarized in Figure 3.1 as a block diagram.

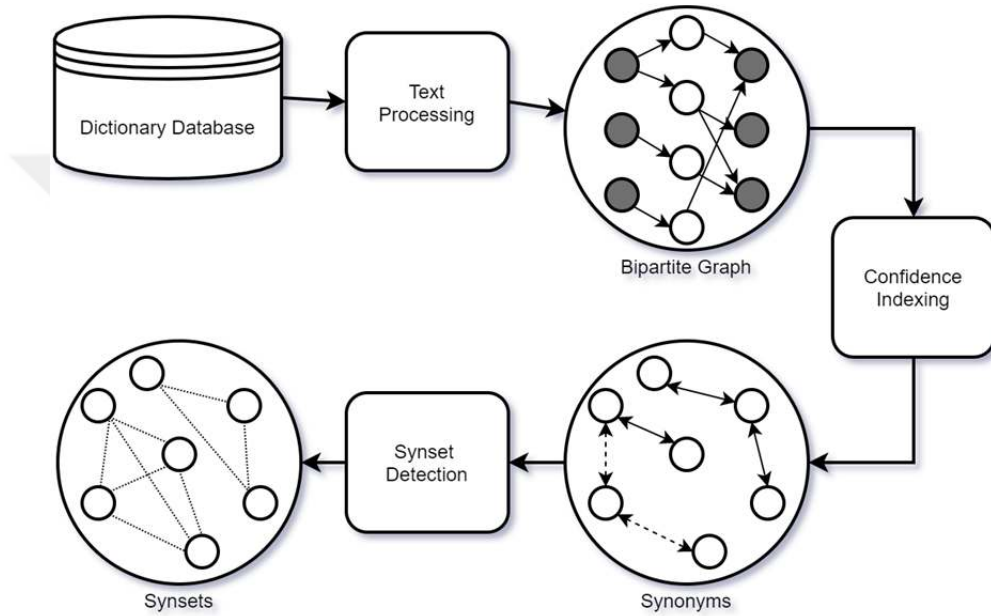


Figure 3.1. Summary of the study

Dictionary database is loaded with lemmas and their definitions from Contemporary Turkish Dictionary (CTD). Lemmas and definitions from database are processed with text processing methods to create a bipartite graph model of the semantic network. Other relations except synonymy, are extracted at this phase. In confidence indexing process, senses in the bipartite graph are processed with confidence indexing method and labeled with index values in the semantic network. Spanning tree-based synset detection applied on indexed synonyms to find synsets in the semantic network.

Semantic networks are prepared using lexical sources as dictionaries, encyclopedias, text corpora or online text mining from the internet. In this study, a Turkish dictionary, Contemporary Turkish Dictionary (CTD), from Turkish Language Association (TLA) is used to create the semantic network for Turkish. Before explaining the data used, it is useful to highlight some important details about dictionary preparation methodology and architecture. Dictionaries are generally designed and prepared according to some fundamental definition rules proposed by linguistics (Jackson 2002). If these rules are applied correctly and consistently in a dictionary, a machine-readable dictionary, in which semantic relation parsing from definitions is automatically possible, is obtained. On the other hand, even if only some of these rules are not followed, the validity of logical architecture of the dictionary can be compromised. Thus, serious semantic errors may arise in the relations to be obtained by automatic text processing.

3.1. Dictionary Data And Preprocessing

CTD is the oldest contemporary Turkish dictionary which is still on print with its eleventh edition. This dictionary, which started in the 1940s, has undergone many updates since its first edition, and the online version has been made public with the rise of internet era. In this study, the online version of CTD is used with text processing. Lemmas in CTD is stored in a flexible data structure due to having varied number of definitions, compound words and phrases.

A NoSQL database is used to store lexical data with JSON format for the subsequent text processing tasks. During the parsing process, homonymy, part of speech, structure of the word, term information and irregularities about suffixes are also extracted for lemmas and their senses. In Figure 3.2, output of lemma “öğrenci” (student) from online CTD and in Figure 3.3, record of lemma “öğrenci” in NoSQL database is presented with its document structure. As a result, a database containing a total of 91,363 lemmas with 121,357 definitions is created.

öğrenci 

1. *isim* Öğrenim görmek amacıyla ders alan kimse, okul çocuğu, talebe, şakirt.
2. *isim* Bir bilim veya sanat yetkilisinin gözetimi ve yol göstericiliği altında belli bir konuda çalışan kimse:
Kant'ın öğrencisi.
3. *isim* Özel ders alan kimse.

Birleşik Kelimeler

öğrenci belgesi	öğrenci bileti	öğrenci kimliği	öğrenci yurdu	ekstern öğrenci
-----------------	----------------	-----------------	---------------	-----------------

Figure 3.2. Output of lemma "öğrenci" (student) from Online CTD

```

_id: ObjectId("58148e7ad02c0a1ffc9b795d")
Name: "öğrenci"
Entry: "öğrenci"
Homonym: 1
Expression: " "
Inflection: ""
Language: " "
Reference: "- "
State: "güncel"
Structure: "- "
Type: "isim"
Usage: " "
~ Definitions: Array
  ~ 0: Object
    Sense: 1
    Type: "isim"
    Term: " "
    Objective: " "
    Explanation: "Öğrenim görmek amacıyla ders alan kimse, okul çocuğu, talebe, şakirt"
  ~ 1: Object
    Sense: 2
    Type: "isim"
    Term: " "
    Objective: " "
    Explanation: "Bir bilim veya sanat yetkilisinin gözetimi ve yol göstericiliği altınd..."
  ~ 2: Object
    Sense: 3
    Type: "isim"
    Term: " "
    Objective: " "
    Explanation: "Özel ders alan kimse"
~ Sentences: Array
  ~ 0: Object
    Sense: 2
    Text: "Kant'ın öğrencisi."
    Source: "- "
~ Idioms: Array
~ Compounds: Array
  ~ 0: Object
    Text: "öğrenci belgesi"
  ~ 1: Object
    Text: "öğrenci bileti"
  ~ 2: Object
    Text: "öğrenci kimliği"
  ~ 3: Object
    Text: "öğrenci yurdu"
  ~ 4: Object
    Text: "ekstern öğrenci"

```

Figure 3.3. Record of lemma "öğrenci" (student) in NoSQL database

CTD does have both single-word lemmas and multi-word expressions (MWEs). Single-word lemmas can be in forms of different word structure for example; “göz” (*eye*) as a root word, “gözlük” (*eyeglasses*) as a derived word or a compound word such as “gözyaşı” (*tears*). Multi-word expressions can be compounds, idioms and proverbs such as “göz akı” (*sclera*), “göz atmak” (*take a glance at*) and “gözden irak olan gönülden de irak olur” (*out of sight, out of mind*), respectively. Compounds, idioms and proverbs are referenced in the lemma of words that form them, in the CTD dictionary. And MWEs are essential for the semantic network because they can be a semantical bridge between yet unrelated single-word lemmas. On the other hand, derived words are not clearly defined in lemma properties, because native speakers of Turkish can perform stemming according to the definition of a word. And the lack of this property prevents a proper morpho-semantic analysis with machine on dictionary data.

Polysemy is another important case which is essential in the semantic network analysis for sense disambiguation. Synonym, antonym and other semantic relations depends on the disambiguation of senses and their semantic relations to other senses. For example, lemma “almak” (*to take*) has 33 senses while lemma “vermek” (*to give*) has 22 senses. These two verbs are existed with high frequency in Turkish texts thus it is vital to solve the problem of finding the correct sense for such polysemous verbs. Statistical numbers about polysemy in the lemmas and their senses, are shown in Table 3.1.

Table 3.1 Statistics about Polysemous and Univocal Lemmas

	Univocal		Polysemous		Total	
	Lemma	Sense	Lemma	Sense	Lemma	Sense
Single-word	46,582	46,582	14,344	39,617	60,926	86,199
MWE	26,553	26,553	3,884	8,605	30,437	35,158
Total	73,135	73,135	18,228	48,222	91.363	121.357

In Table 3.1, univocal columns show the number of single-word lemmas and MWEs which does only have one sense, and the polysemous columns show the

number of lemmas with multiple senses and their total number of senses. While approximately 24% of single-word lemmas are polysemous, this rate drops to 13% in MWEs. Also average sense numbers of polysemous single-word lemmas are greater than the the average sense numbers of polysemous MWEs. Looking at the results of these two numerical comparisons between single-word lemmas and MWEs, single-word lemmas are more likely to be polysemous.

In Table 3.2, distribution of sense numbers of lemmas are listed according to single-word lemmas and MWEs. If the number of sense of a lemma increases, the probability of that lemma being a single-word lemma also increases. A single-word lemma have maximum senses up to 56 senses while MWEs have senses up to 9 senses.

Table 3.2 Sense numbers of Single-word Lemmas and MWEs

Senses	Single-word Lemmas		MWEs		Total
	Counts	Percentage	Counts	Percentage	
1	46,582	63,7	26,553	36,3	73,135
2	18,744	74,3	6,474	25,7	25,218
3	8,304	84,7	1,503	15,3	9,807
4	4,112	89,4	488	10,6	4,600
5	2,325	97,5	60	2,5	2,385
6	1,560	97,0	48	3,0	1,608
7	994	99,3	7	0,7	1,001
8	672	97,7	16	2,3	688
9	675	98,7	9	1,3	684
10+	2,231	100,0	0	0,0	2,231

In the processed dataset, there is a property list for lemmas, and polysemous lemmas with multiple senses may have different property values for each sense. Beside the property list of lemmas, each sense has own property list to keep different values from lemma's property list. Table 3.3 shows the property list of a lemma with their definitions.

Table 3.3 Property list of a lemma

Property	Essential	Definition
Name	Yes	Name of the lemma
Entry	Yes	Entry name of the lemma with homonym order
Homonym	Yes	Homonym order of the lemma
Expression	No	Expression form of the lemma
Inflection	No	Inflection form of the irregular lemma
Language	No	The origin language of the loan word lemma
Type	No	Part of speech type of the lemma
Usage	No	Type of speech where the lemma used
State	No	Actuality status of the lemma
Structure	No	Word structure of the lemma
Reference	No	Reference information of the lemma

The reference property of a lemma references to the original lemma if it is a misused version of the original lemma or references to the equivalent lemma in Turkish if it is a loan word. If the *Reference* property of a lemma exists lemma does not have any sense and lemma is not suitable for the semantic network structure. These only-referencing lemmas are excluded from the semantic network.

Lemmas may have various properties, some of these properties are essential for the lemma and some of them are additional information which is added to the lemma if needed. The *Name* property of a lemma, represents the plain name of the lemma. *Entry* property of the lemma represents the name of the lemma written in the dictionary as an entry. *Entry* property value is identical to *Name* property value if the lemma is not a homonym. Else, *Entry* property represents the plain name of the lemma with homonym order. The *Homonym* property of the lemma shows the homonym order of the lemma and if a lemma is not homonym lemma, then its default *Homonym* property value is 1 in the dataset. *Name* and *Homonym* properties together form the primary key for the dataset. In Figure 3.1, two lemmas are shown with the data structure from NoSQL database.

<pre> _id: ObjectId("58148dfad02c0a1ffc9b7442") Name: "ev" Entry: "ev" Homonym: 1 Expression: " " Inflection: "" Language: " " Reference: "-" State: "güncel" Structure: "-" Type: "isim" Usage: " " Definitions: Array > 0: Object > 1: Object > 2: Object > 3: Object Sense: 4 Type: "isim" Term: " " Objective: " " Explanation: "Soy, nesil" Sentences: Array Idioms: Array Compounds: Array </pre>	<pre> _id: ObjectId("5814ae02d02c0a1ffc9c6212") Name: "kurt" Entry: "kurt (I)" Homonym: 1 Expression: " " Inflection: "-du" Language: " " Reference: "-" State: "güncel" Structure: "-" Type: "isim" Usage: " " Definitions: Array > 0: Object > 1: Object Sense: 2 Type: "isim" Term: " " Objective: " " Explanation: "Bir yeri, bir şeyi iyi bilen" > 2: Object Sentences: Array Idioms: Array Compounds: Array </pre>
---	---

Figure 3.4. NoSQL data structure of Lemmas "ev" and "kurt"

Lemma "ev" (*house*) has same identical value for *Name* and *Entry* property and default value for *Homonym* property, while homonymous lemma "kurt" (*wolf*) has different *Name* and *Entry* property and value of *Homonym* property is parsed from *Entry* property. *Reference* properties are obsolete with "-" mark. As seen in the Figure 3.1, there are arrays of definitions, sentences, idioms and compounds in the structure of a lemma, they will be explained after the properties of the lemma. *Expression* property of the lemma is one of the additional information described in the dictionary. This property explains the way that the lemma expressed in use. This property exists in very few lemmas, although it is a semantically valuable tag for the lemmas. In Table 3.4, all enumerated *Expression* property values are listed. *Metaphor* tag in *Expression* property, denotes that the lemma is a natural metaphor and it is all senses are also metaphor. *Offensive*, *Humorous* and *Ridiculous* tags denote that the lemma is used to intent these expressions. Finally, *Slang* tag denotes that the lemma is a slang word.

Table 3.4. Enumerated values of Expression property of the lemma

Expression	Explanation	Count
Metaphor	Using metaphorically	1,296
Offensive	Using to insult	26
Humorous	Using to joke	47
Ridiculous	Using to ridicule	59
Argo	Slang lemma	436

Inflection property in a lemma shows the irregular suffix pattern that a word takes according to vowel and consonant events in Turkish natural language. Turkish vowel harmony rules are applied to Turkish words effortlessly. On the other hand, loan words can be irregular when try to add suffixes according to vowel and consonant events. These suffix patterns describe how to apply the suffix for loan words in Turkish. If there is no suffix pattern described, suffix is applied to the lemma by the rules. There are 11,800 unique lemmas form at least one irregular suffix pattern.

Irregular suffix pattern list, "-ir, -ür, -ar, -ır, -dar, -ur, -er, -der", are for the verbs take third singular person (3SG) simple present tense suffix. There are 167 verb lemmas have a irregular 3SG simple present tense suffix such as “*yenmek*” (to be eaten).

Since 1930s, Turkish alphabet is based on Latin script with highly phonetic notation. Loan words can be dissonant to Turkish palatal harmony, when having different pronunciation than its written form. Irregular suffix pattern list, "-fî, -li, -mi, -kî, -lî, -lû, -ri, -fî" is necessary when a loan word such as “*golf-golfü*” (golf) is dissonant to Turkish palatal harmony.

Table 3.5 shows the irregular suffix pattern list when a lemma takes a suffix while vowel drop occurs. Vowel drops can occur in two conditions when a lemma takes a suffix. First case that causes is the words which represents body parts take a suffix such as “*alın - alnı*” (forehead). And the second case is the loan words from Arabic such as “*zihin – zihni*” (mind). There are 144 irregular suffix patterns for the vowel drop in Turkish.

Table 3.5. Irregular suffix patterns for vowel drop of Inflection property

Irregular Suffix Patterns									
-bli	-fsi	-hni	-kki	-lmü	-rmü	-stı	-tmi	-ydi	-zki
-brı	-fyi	-hri	-kki	-lnı	-rnı	-şfi	-tnı	-ydu	-zli
-bri	-fzı	-hri	-kli	-lsü	-rnu	-şmı	-tni	-yfi	-zmi
-bzı	-ğlu	-hrü	-kli	-mli	-sbı	-şrı	-trı	-yfi	-zmi
-cmi	-ğni	-hsı	-kmü	-mri	-sci	-şrı	-tri	-yli	-zni
-cri	-ğrı	-hsi	-knü	-mrü	-sfı	-şrü	-tru	-ynı	-znü
-cvi	-ğru	-hşu	-kri	-msi	-shi	-şvi	-ulu	-yni	-zri
-czi	-ğrü	-hvi	-krü	-mzi	-slı	-şyi	-uzu	-ynu	-zrü
-çhi	-ğsü	-hvi	-ksi	-mzu	-sli	-tbu	-vci	-yri	-zvu
-dli	-ğvı	-hyi	-kşı	-nlü	-slü	-tfı	-vli	-yri	
-dri	-ğzı	-hzi	-kti	-nni	-smı	-tfu	-vmi	-yti	
-dri	-ğzu	-kdi	-kzi	-nzi	-smi	-thı	-vri	-yzi	
-fku	-hdi	-kfi	-kzi	-psi	-snü	-thi	-vri	-yzi	
-fni	-hli	-khi	-lfü	-ptı	-srı	-tku	-vr'i	-zbi	
-frü	-hmi	-kı	-lmi	-rmi	-sri	-tli	-ybi	-zfi	

Loan words from Arabic which have twin consonants at the end, stripped single consonant at the end when borrowed. This condition reverts back when a suffix or an auxiliary verb combined with the loan word such as “*af - affi*” (*amnesty*). Table 3.6 shows the irregular suffix pattern list for the loan words from Arabic which have twin consonants.

Table 3.6. Irregular Suffix Patterns for loan words with twin consonants

Irregular Suffix Patterns							
-bbı	-ddı	-ffi	-lli	-mmi	-rrı	-ssi	-yyı
-bb'i	-ddi	-hhi	-llü	-nnı	-rri	-ssü	-zzi
-cci	-ffi	-kk'ı	-mmı	-nnü	-ssi	-tti	

Table 3.7 shows irregular suffix pattern list that suffixes can be formed when words and other suffixes need connecting sounds to help the consonance of the Turkish language. These connecting suffixes act like a glue to connect words

and suffixes but they do not have a semantic meaning at all such as “*su – suyu*” (*water*).

Table 3.7. Irregular Suffix Patterns for Suffix with Connecting Sounds

Irregular Suffix Patterns				
-nı	-si	-yı	-yu	-yü

Table 3.8 shows irregular suffix pattern list that suffixes can be formed when consonant assimilation occurs. Consonant assimilation is a sound event occurs when a word that ends with a stop consonants like “p, ç, t, k” takes a suffix beginning with a vowel. And “p, ç, t, k” consonants change to “b, c, d, (g, ğ)” consonants to keep the harmony of the lemma's pronunciation such as “*dört – dördü*” (*four*).

Table 3.8. Irregular Suffix Patterns for Suffix with Consonant Assimilation

Irregular Suffix Patterns						
-dü	-ku	-t'u	-cdı	-ti	-k'ü	-gu
-ç'u	-k'u	-t'i	-ğı	-ğu	-bı	-gi
-cı	-k'i	-ç'ı	-dı	-bu	-cü	-ğü
-t'ü	-gü	-p'i	-du	-ğı	-t'ı	-çı
-dı	-p'ü	-p'u	-ci	-pı	-k'ı	-bü
-ki	-gı	-bı	-cu	-p'ı	-ç'ı	

Language property defines the origin of the word, and this property can have null value which point outs that its origin is Turkish. In this property there can be more than one natural language described which means that the word is a compound word combined from at least two different natural languages.

This condition occurs with Arabic and Persian words due to the common history of Turkish and these languages. Table 3.9 shows the numbers of lemmas that have a origin from foreign languages.

Turkish language is an agglutinative language and deriving new words using suffixes is the main source of new words in Turkish. When a loan word is

combined with a suffix to create a new word, the newly created word assumed a Turkish word in CTD dictionary.

Table 3.9. Loan words numbers with their origin language

Language	Count	Language	Count	Language	Count
Arabic	6644	Russian	46	Sanskrit	8
French	5750	Spanish	35	Sogdian	5
Persian	1867	Armenian	23	Slavic	4
Italian	661	Bulgarian	20	Tibetan	3
English	575	Hungarian	16	Chinese	3
Romaic	497	Japanese	15	Indian	3
German	118	Mongolian	13	Portuguese	3
Latin	70	Serbian	10	Roman	1
Greek	49	Hebrew	9		

The numbers shown in Table 3.9, includes lemma "*mevsim*" (*season from Arabic*) but does not count "*mevsimsel*" (*seasonal*). This is the main reason why Turkish has irregular suffix patterns above the expected numbers. As the interaction of Turkish with other languages increases, especially with borrowing words in science, art and technology, new cases will emerge for these irregular suffix patterns.

State property defines that if the word is in active use in Turkish language or it is an old, unused or discarded word with two terms: *güncel* (*actual*) and *eskimiş* (*obsolete*). In the database there are 87,230 words in current usage and 4,310 words are abandoned.

Structure property defines the structure of the word, it has three class types: *basic*, *derived* and *compound* but only compounds are defined with the explicit data from CTD dictionary, *basic* and *derived* values was defined with "-" mark in the parsing process because of insufficient data. After extracting derived words from the dataset, derived lemmas is updated with proper tag. Basic word analysis is excluded from this study, as it requires intensive manual etymological observation by experts.

Type property defines the type of the lemma, it has 10 Part of Speech (POS) types and 4 other types. POS types can vary in different languages and Turkish has *nouns* and *proper nouns*, *adjectives*, *adverbs*, *verbs* and *auxiliary verbs* as content words, and *pronouns*, *conjunctions*, *prepositions* and *exclamations* as function words.

Beside these POS types, there are lemmas without *Type* property defined which are later defined as *letters*, *element signs*, *abbreviations* and *phrases*. POS tags of phrases are not explicitly defined in the lemma properties. In semantic relation analysis, POS tags must be compared to relate two different lemmas or their senses if they have the same POS tags.

Phrases with no POS tags are analyzed, and if they are compound verbs made with auxiliary verbs than tagged with "verb" tag, else they are tagged with "phrase" tag. In Table 3.10, *Type* property values, POS types and other types, shown with numbers.

Table 3.10. Type property values with their frequency in the dataset

Type	Count	Type	Count	Type	Count
Noun	50,514	Proper Noun	2,025	Adjective	11,216
Adverb	2,414	Verb	20,984	Auxiliary Verb	4
Pronoun	78	Conjunction	37	Preposition	38
Exclamation	178	Letters	37	Element Signs	102
Abbreviation	1	Phrase	3,735		

Usage property describes the style of speech that the lemma is used. This property has three class types: *colloquial language*, *vulgar language* and *Informal language*.

Colloquial indicates that the word is used in ordinary conversation, *vulgar* indicates that the word is used in offensive and obscene language and *informal* indicates that word is used in a familiar and unofficial conversation.

In Table 3.11, style of speech class types are shown with the numbers in single-word lemmas and MWEs. As seen in Table 3.11, vulgar and informal language style tagged lemmas exist in MWEs more than in single-word lemmas.

Table 3.11. Usage property values with their numbers in the dataset

Style of Speech	Count	
	Single-Word Lemma	MWE
Coloquial Language	1,419	257
Vulgar Language	35	52
Informal Language	64	127

Definitions list is the first inner array element of the lemma which contains definitions of the lemma. Lemmas have at least one definition which is the literal meaning of the lemma. If a lemma is polysemous, beside the denotation, it can have connotations or figurative expressions as definitions. A definition is not explicitly described as the denotation or a connotation in CTD dictionary. In Figure 3.5, lemma “alet” (*tool*) has 4 senses. First sense is denotation while the second sense is a connotation. Third sense of the lemma is technical term and the last sense is a metaphor.

alet  (a:let), Arapça âlet 1. <i>isim</i> Bir el işini veya mekanik bir işi gerçekleştirmek için özel olarak yapılmış nesne. 2. <i>isim</i> Bir sanatı yapmaya, uygulamaya yarayan özel araç. 3. <i>isim, teknik</i> Bir makineyi oluşturan ve işlemesine yardım eden parçalardan her biri. 4. <i>isim, mecaz</i> Maşa: "Birtakım teşebbüslerini gerçekleştirmesi yolunda onu bir alet gibi kullanıyor." - Yakup Kadri Karaosmanoğlu
--

Figure 3.5. Output of lemma "alet" (*tool*) from online CTD

Each definition of a lemma has five properties. *Sense* property is the order of the definition according to CTD dictionary. *Type* property is the type of the sense, this property can be different from the lemma's general *Type* property because a Turkish adjective can be used as a noun or an adverb in the context without a morphological change. If the definition indicates a term for a discipline, *Term* property describes the discipline of the term. There are 37 discipline class types enumerated for *Term* property (Table 3.12).

Table 3.12. Diciplines class types of Term property

Discipline	Count	Discipline	Count	Discipline	Count
botany	1,890	theology	469	mineralogy	138
zoology	1,494	anatomy	464	logic	117
chemistry	927	literature	420	theater	103
medicine	842	economy	369	Informatics	89
physics	772	biology	337	pedagogy	79
grammar	733	psychology	317	meteorology	51
philosophy	730	geography	316	mining	46
sports	716	astronomy	315	geometry	27
marine	619	sociology	287	physiology	22
mathematics	605	business	277		
law	598	geology	245		
history	544	architecture	162		
music	539	cinematography	161		
military	505	technics	142		

Objective property is an operative property only for verbs in the database. If the lemma is a transitive verb than this property describes the suffixes that objects takes when used with this transitive verb such as “-e gitmek – ev(e) gitmek” (to go - to go home). If it is an intransitive verb than it has a “nsz” tag which is an abbreviation for Turkish “nesnesiz” meaning without an object such as “uyumak” (to sleep). Senses of a verb may have different transitivity according to their meanings.

A transitive sense of a polysemous verb has at least one suffix to be appended on object. Synonym verbs must have the same transitivity state and

suffix. In Table 3.13, number of *Objective* property classes for the verb senses in the database. There are total 29,540 verb senses in the database, but as seen in Table 3.13 only 15,824 intransitive and transitive tags used to describe transitivity for the verb senses. Also some the verb senses do have more than one transitivity tag. *Objective* property is not applicable to validate synonym relations between two verbs because the lack of *Objective* property in verb senses of MWEs in CTD.

Table 3.13. Objective property tags

Tag	Definition	Count
nsz	no object(intransitive)	6,976
-e	to object	2,244
-i	the object	7,722
-le	with object	457
-den	from object	445

Explanation property contains the definition of that sense in the *Definition* list. This content is the main source for the semantic network design in further text processing in this study. After this aggregation process, n-gram for each word applied according to this property with word-based n-gram analysis.

Sentences list is the second inner array element of the lemma which contains exemplary sentences for definitions of the lemma. Each sentence element has three properties. *Sense* property indicates the sense order that exemplary sentence belongs. *Text* property has the sentence content and *Source* property indicates the source of the sentence which can be an author, poet, politician etc.

Idioms list is the third inner array element of the lemma which contains every idiom that contains the lemma itself. This element has only one property: *Text* which describes the idiom contains the lemma.

Compounds list is the fourth inner array element of the lemma which contains every compound that formed by the lemma. This element has only one property: *Text* which describes the compound contains the lemma.

These are the data acquired from the TLA Contemporary Turkish Dictionary after some parsing problems pruned and cleaned. When the dictionary data is analyzed in terms of the fundamental definition rules (Jackson 2002), the following notes are taken:

- The definitions do not contain label of meaning type except metaphorical meaning. However, a hidden ranking architecture is observed (1. denotation, 2. connotation, and if any 3. metaphorical meaning).
- Part of Speech (POS) tags are given in most of the lemmas. In only some definitions under the lemma, if its label is different from the lemma, another POS tag is observed.
- The definitions begin with the explanation prepared with its hypernym, and/or synonyms separated by commas.
- For Turkish, synonym relationships occur with polysemy or borrowing situations (Karaağaç 2013).

The deficiencies and problems that emerged in because of the structural conditions of the Contemporary Turkish Dictionary data above are listed below:

- Lemmas in MWE format do not have any POS tags.
- Some synonym relations are defined between the senses with different POS tags.
- There is no notation about the derivational suffixes used in the derived lemmas.
- Synonym relations in the metaphorical definitions usually do not reference back.

In the study, the absence of a POS tag in MWEs is fixed by assigning the POS tag of the last word in the MWE. Besides, POS tag matching is also considered in the detection of semantic relationships. Although Turkish is a

language that is included in the group of agglutinative languages, the fact that the derivational suffix details are not presented caused the morpho-semantic connection to be unavailable in semantic analysis. The new lemmas derived from a polysemous root lemma in Turkish may cause polysemy to reveal or increase complex connections for synonymy and semantic distance (Zhu 2014).

Problems in semantics may not be solved by algorithms due to incorrect definitions when processing text. Words in hierarchical order must be defined carefully for feature and hierarchy extraction. For example, word “eşya” (*goods, stuff*) has a definition that ends with “... cansız nesneler“ (... *inanimate objects*) that references to its hypernym “nesne” (*object*). And the definition of “nesne” references to “... cansız varlık, şey, obje” (... *inanimate being, thing, object*). Lemma “nesne” already has a definition that indicates it is an inanimate being so lemma “eşya” definition does not need the “inanimate” attribute because of inheritance. Another example for semantic problems, “kedigiller” (*felines*) word has an explanation: “Kedi, aslan, kaplan, pars vb. hayvanları içine alan etçil memeli hayvanlar sınıfı” which says that animal *class* of carnivore mammals that includes animals like cat, lion, tiger, leopard etc. On the other hand, the explanation for “köpekgiller” (*canines*) is “Köpek, kurt, çakal, tilki vb. etobur memelileri içine alan hayvan familyası” which says that animal *family* of carnivore mammals that includes animals like dog, wolf, jackal, fox etc. These two definitions ends with “sınıfı” (*class of*) and “familyası” (*family of*) words which must be synonym for each other according to the biological classification. But in the definitions of “sınıf” and “familya”, there is no synonym relation between these words. In fact, despite the definitions of CTD, felines and canines are described as subfamily in the biological classification hierarchy.

3.1.1. N-Gram Analysis

Explanations in the database can have distinguishable patterns for some words or word groups which can be useful to analyze relationships between words and to find forms of derivative suffix with definition patterns. In this study, an

application called N-Gram Analyzer designed to analyze explanations for word-based n-gram. This tool has console based and GUI based two running modes. Console mode of the program is an automatic n-gram extractor for the dictionary dataset. When the program started in console mode, it connects to the NoSQL database and begin to query for each definition of lemmas to extract the word-based n-grams. Before extracting n-grams in the definition, all the numbers, punctuations and extra white spaces are pruned from definitions. After n-gram extraction, n-grams are found from 1-gram to the longest n-gram, 56-gram. And when the extraction task finished program stores all n-grams to NoSQL database and exits to operating system. As seen in the Figure 3.2, there are four properties except predefined primary key *_id*, and one inner array element called *Data* for each definition contains the n-gram.

Key	Value	Type
▼ (1) ObjectId("5c818aef7c65dc764a46e3a3")	{ 6 fields }	Object
_id	ObjectId("5c818aef7c65dc764a46e3a3")	ObjectId
# NGramSize	1	Int32
# NGramID	6580	Int32
"" Term	üniversite	String
# Count	9	Int32
▼ Data	[9 elements]	Array
> [0]	{ 4 fields }	Object
> [1]	{ 4 fields }	Object
> [2]	{ 4 fields }	Object
▼ [3]	{ 4 fields }	Object
"" Name	darülfünun	String
# Homonym	1	Int32
# Sense	1	Int32
"" Definition	üniversite	String
> [4]	{ 4 fields }	Object
> [5]	{ 4 fields }	Object
> [6]	{ 4 fields }	Object
> [7]	{ 4 fields }	Object
> [8]	{ 4 fields }	Object

Figure 3.6. Document structure of n-gram data in NoSQL database

NGramSize property indicates number "n" for the n-gram, *NGramID* property indicates the unique ID of the n-gram, *Term* property contains the n-gram text which is generated from the definition, and *Count* property is the n-gram's frequency of appearing in the definitions. An n-gram may appear more than one in a definition so each of these appearances will be counted. Inner array element *Data* has all the appearing senses of the n-gram indicated in the *Count* property. Each element in *Data* list has four properties. *Name* property indicates the name of the lemma. *Homonym* property indicates the homonym order of the lemma. *Sense* property indicates the order of the sense of the lemma and last property *Definition* has the explanation of the lemma according to the *Sense* order. This data model holds all data needed for analyzing the dictionary for a complete word-based n-gram processing. In Table 3.14, distribution of n-grams are shown from 1-gram to 56-gram. 3-gram, 2-gram and 4-gram are the largest n-grams with most useful patterns for semantic relation extracting.

Table 3.14. N-Gram Distribution for the definitions from CTD dataset

N-Gram	Count	N-Gram	Count	N-Gram	Count
1	80716	20	7366	39	69
2	324944	21	5712	40	52
3	356159	22	4430	41	41
4	307899	23	3440	42	32
5	255493	24	2680	43	25
6	207165	25	2078	44	22
7	167740	26	1614	45	19
8	135193	27	1253	46	16
9	108414	28	986	47	14
10	86616	29	775	48	12
11	68912	30	608	49	10
12	54561	31	474	50	8
13	42998	32	376	51	6
14	33739	33	302	52	5
15	26354	34	242	53	4
16	20526	35	189	54	3
17	15897	36	148	55	2
18	12310	37	116	56	1
19	9522	38	89		

NGram Analyzer tool in GUI mode, has two functions to view n-grams from the dictionary data. First function can load all the n-grams according to minimum *Count* and *n* values as a list. This function is useful to find significant patterns that can be used for semantic relation extracting. And when results listed in NGram List table, user can select any of these ngrams to view definitions in the NGram Data List table. When an n-gram selected in the NGram List table, program calculate the Maximum Likelihood Estimation for the n-gram while loading the NGram Data. And the second function is an n-gram search with a text input. After a search, if the text input exists as an n-gram it will be listed on NGram List table. Maximum likelihood estimation (MLE) is a practical approach to estimate probabilities which can be easily applied for n-gram analysis on a corpus as shown in the Equation 3.1(Jurafsky and Martin, 2008).

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}w)} \quad (3.1)$$

Probability of a bigram " $w_{n-1} w_n$ " words can be estimated with count of " $w_{n-1} w_n$ " words $C(w_{n-1}w_n)$ by $C(w_{n-1})$, the count of word w_{n-1} , to find the all bigrams begins with word w_{n-1} . And for a trigram " $w_{n-2} w_{n-1} w_n$ " words can be estimated with count of " $w_{n-2} w_{n-1} w_n$ " words $C(w_{n-2} w_{n-1} w_n)$ by $C(w_{n-2} w_{n-1})$, the count of bigram " $w_{n-2} w_{n-1}$ ". The Equation 3.1 can be simplified with this approach to the Equation 3.2.

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})} \quad (3.2)$$

As seen in the Figure 3.4, a bigram "*olma durumu*" (*state of being*) is searched in NGram Analyzer's GUI mode. And the bigram is choosen in the NGram List table. The MLE value of the bigram is calculated with unigram "*olma*" (*to be*) and the result is 0.9488 which means olmost every word "*olma*"

comes before the word "*durumu*". This analyze can easily lead to the longest proper pattern for the ngrams found in the CTD dataset. And the proper pattern, "*olma durumu*" bigram, is clearly observed in the NGram Data List table.

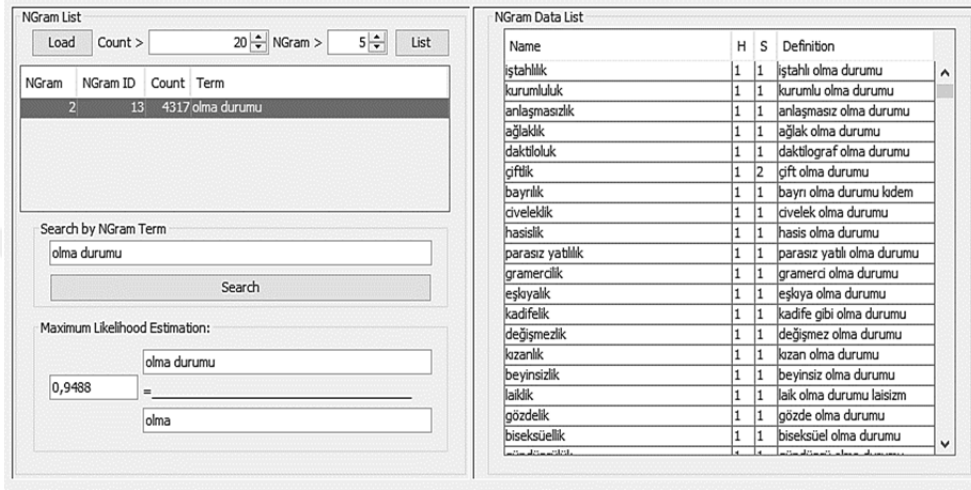


Figure 3.7. Graphical user interface of NGram Analyzer

N-gram database is observed for frequency of the n-grams and 50 is manually choosen as a proper pattern threshold for the frequency of the n-gram to find proper patterns for semantic relations. Every n-gram that has a frequency greater than 50 are taken for manual observation to find a proper pattern of a semantic relation. 1886 n-gram patterns are found with a frequency above 50, however 1535 of them are 1-gram usually insignificant for semantic relation patterns. Top frequencies for [2,3,4,5,6,7]-grams are found 4317, 1778, 1776, 80, 59 and 51, respectively. But top frequencies of 3-grams and 4-grams actually the same pattern. Manual observation is applied to find the longest word pattern of repeating patterns. After manual observation is completed there are several n-gram patterns for *Hypernym* and *Group Of* semantic relations which are the primary semantic relations in semantic networks. And other semantic relations are found that do not exist in semantic networks like WordNet. These semantic relations are

also added to the semantic network to enrich the relations between nodes. Derived words in CTD can combine these relations with *Derive* relations to form morpho-semantic relations in further analysis. In Table 3.15, Group Of patterns are shown with [elementName] and [groupName] tags. A pattern with [elementName] is found in definition of a lemma, if the lemma is a group name. And a pattern with [groupName] denotes that the lemma of that definition is the element of that group. "Bu Türkçe ile yazılmış olan" pattern is found in the definitions of the elements that belong to the Turkish language group.

Table 3.15. Patterns from CTD found by n-gram analysis for Group Of

N-Gram	Pattern	Relation
1	[elementName] topluluğu	[elementName]
1	[elementName] bütününü	[elementName]
1	[elementName] kümesi	[elementName]
1	[elementName] tümü	[elementName]
1	[elementName] sınıfı	[elementName]
1	[elementName] takımı	[elementName]
1	[elementName]+gillerden	[elementName]
4	[groupName] iline bağlı ilçelerden biri	[groupName]
5	Bu Türkçe ile yazılmış olan	Türkçe

In Table 3.16, n-gram patterns for *Hypernym* relations are listed, [hypernym] tag corresponds to the word substituting the hypernym. In linguistic typology, Turkish language word order is Subject-Object-Verb (SOV) and a verbal sentence ends with a verb or a nominal sentence ends with a noun (Comrie 1989). And verb or noun at the end of the sentence has inflectional suffixes for tense, mood and person. However, in a dictionary, generally a definition sentence ends with the verb or noun in lemma form. 1-Gram patterns seen in Table 3.16, are formed with the lemma at the end of the definition sentences. Other n-grams in Table 3.16, are extracted from in different parts of a definition sentence according to their appearance. And "Bu renkte olan" pattern obtained by extracting whole definition sentence.

Table 3.16. Patterns from CTD found by n-gram analysis for Hypernymy

N-Gram	Pattern	Hypernym Lemma
1	yer*	yer (place)
1	madde*	madde (matter)
1	hayvan*	hayvan (animal)
1	bitki*	bitki (plant)
1	bilimi*	bilim (science)
1	yemek*	yemek (food)
1	adı*	ad (name)
1	kadın*	kadın (woman)
1	erkek*	erkek (man)
1	kişi*, kimse*	kimse (person)
1	element*	element (element)
1	simge*	simge (symbol)
1	işi*, iş*	iş (job)
2	bir tür [hypernym]	[hypernym]
2	bir [hypernym] türü	[hypernym]
2	bir [hypernym] tipi	[hypernym]
2	bir familya	familya (family)
2	[hypernym] genel adı	[hypernym]
2	çok yıllık	bitki (plant)
2	kullanılan araç	alet (tool)
2	yarayan araç	alet (tool)
2	kullanılan alet	alet (tool)
2	ses çıkarmak	ses (sound)
2	bir balık	balık (fish)
2	bir ağaç	ağaç (tree)
2	yaşayan bir	canlı (alive)
2	inceleyen bilim	bilim (science)
3	bu renkte olan	renk (color)
3	ölçmeye yarayan alet	alet (tool)
3	bir bitki familyası	familya (family)
3	bir süs bitkisi	bitki (plant)
3	otsu bir bitki	bitki (plant)
3	kimse veya şey	kimse (person), şey (thing)
3	inceleyen bilim dalı	bilim (science)
3	işi veya mesleği	iş (occupation)
3	klasik Türk müziğinde	müzik (music)
3	bir seslenme sözü	söz (saying)
4	anlamında kullanılan bir söz	söz (saying)
4	deyiminde geçen bir söz	söz (saying)
4	sırasında çıkan sesin adı	ses (sound)

In Table 3.17, patterns are shown with other semantic relations. In addition to the *Hypernym* and *Group of* semantic relations, these semantic relations were also found during the n-gram analysis.

Table 3.17. Patterns from CTD found by n-gram analysis

N-Gram	Pattern	Relation
1	durumu, işi	NounFormOf, VerbFormOf
1	çabucak	Quickly
1	olan	Presence, NounFormOf, AdjectiveFormOf
1	olmayan	Absence, NounFormOf, AdjectiveFormOf
1	ilgili	Related
1	becermek	AbleTo
1	davranmak	Behave
1	bilimi	ScienceOf
2	bir biçimde	AsLike
2	duruma getirmek	ToMake
2	olma durumu	ToBe
2	sebepe olmak	ToCause
2	duruma gelmek	ToBecome
2	yaptığı iş	Master
2	işleten kimse	Manager
2	yapan kimse	Maker
2	satan kimse	Seller
3	yanlışı olan	Supporter
3	halkından olan kimse	From
2	görevli kimse	Responsible
2	gücü yetmek	Ability
3	işine konu olmak	ObjectOf
4	bu dille yazılmış olan	WrittenWith
2	duruma getirmek	BringTo
3	işi veya mesleği	Occupation
2	Bu _____ yapılan	MadeWith
2	işini yaptırmak	Causative
4	iline bağlı ilçelerden biri	Instance

NounFormOf, *AdjectiveFormOf*, *VerbFormOf* semantic relations are the relations between the same concept in different POS tags. *NounFormOf* and *VerbFormOf* relations are asymmetric relations where a concept's noun form and verb form are connected to each other. As an example, "*boyama*" (*noun form* -

painting) and "*boyamak*" (*verb form - "to paint"*) are connected to each other with *NounFormOf-VerbFormOf* relations. However, this situation should not be confused with the concept having identical written form for noun and verb like "*boya*" (*paint*) and "*boya-mak*" (*to paint*).

NounFormOf and *AdjectiveFormOf* relations are asymmetric relations emerged from derivative suffixes "*-li* (*-li, li, lu, lü*)" and "*-siz* (*-siz, -siz, -suz, -süz*)". Derivative suffix "*-li*" converts a noun to an adjective with a meaning of available, existing or with it while derivative suffix "*-siz*" converts a noun to an adjective with a meaning of unavailable, non-existing and without. These "*-li*" and "*-siz*" suffixes also has semantic relations, *Presence* and *Absence* relations, respectively.

Master, Manager, Maker, Seller, Supporter and *Responsible* semantic relations are the relations for a person explicitly explained by their names. As an example a *Maker* relation is created between "*mobilyacı*" (*furnisher*) and "*mobilya*" (*furniture*).

Quickly, AbleTo, Behave, ToMake, ToBe, ToCause, ToBecome, Ability, ObjectOf, BringTo and *Causative* relations are created between a noun form of a verb to a derived version of that verb. For example *Causative* relation created from "*dağıtma*" (*distributing*) to "*dağıttirmek*" (*to make someone to distribute*).

And *Instance* relations are created between a concept and its proper noun instances like "*ilçe*" (*county*) and "*Seyhan*" (*a county in Adana city, Turkey*).

3.1.2. Analysis of Derived and Antonym Words

Derived words are of great importance for the Turkish language. As a member of agglutinative languages, Turkish, evolves generally by generating derived words. Turkish language has approximately 300 inflectional and derivational affixes (Çotuksöken 2011). Some affixes have same written form with different functions. And a derivational suffix can be used in different meanings

according to word's domain. As an example, derivational suffix "-*ci* (-*ci*, -*ci*, -*cu*, -*cü*, -*çl*, -*çl*, -*çu*, -*çü*)" able to generate 13 different concepts according to the word it appended (Zülfikâr 2011). It is one of the active derivational suffixes of Turkish language, easily appended to Turkish or loan words. In CTD, "*televizyoncu*" derived from "*televizyon*" (*télévision from French*) have three different senses: the person who sells television, the person who repairs television and the person who works in a television channel.

It is a compelling task to find morpho-semantic relations between the root lemma and its derivative words in semantic perspective. In this study, guiding rule is accepted as creating semantic relations based on sense level thus derivative relations are searched with the use of definitions. On trying to detect a derivative word, there are some difficulties arise. The first step is the determination of the suffix appended to the derived word. This task can be solved by using a stemming method to reach the root lemma. An attempt is made to find root lemma with the Zemberek API, but it was produced generally multiple results for root lemma because a derived lemma can be generated by combinations of derivative suffixes. And choosing the right lemma from root lemma candidates is a more challenging situation than stemming. A basic stemming method, lemma sampler, is proposed and used in place of Zemberek API for stemming by using the definition of the derived word and total lemma list of CTD. Lemma sampler designed to find longest root lemma from a derived word lemma. The lemma sampler's getLongestRoot procedure's pseudo-code is shown in below:

ALGORITHM 1: Longest Root Method

process getLongestRoot(*Word*, *minWL*)

Roots = empty // empty root list

derived = Word.toLowerCase() // for Turkish İ -> ı and İ̇ -> i

stem = derived

```

continueParsing = true
if derived exist in CTD.Lemmas then
    Roots.add(derived)
else
while continueParsing is true do
    if stem.length > minWL then
        stem = stem.substring(0, stem.length-1)
        if stem exist in CTD.Lemmas then
            Roots.add(stem)
            continueParsing =false
        end
        if stem+"mek" exist in CTD.Lemmas then
            Roots.add(stem+"mek")
            continueParsing =false
        end
        else if stem+"mak" exist in CTD.Lemmas then
            Roots.add(stem+"mek")
            continueParsing =false
        end
    end
end

if Roots.empty then
    continueParsing = true
    stem = derived
    while continueParsing is true do
        if stem.length > minWL then
            stem = stem.substring(0, stem.length-1)
            stem = checkForVowelReduction(stem)

```

```

    stem = checkForConsonantSoftening(stem)
    if stem exist in CTD.Lemmas then
        Roots.add(stem)
        continueParsing =false
    end
    if stem+"mek" exist in CTD.Lemmas then
        Roots.add(stem+"mek")
        continueParsing =false
    else if stem+"mak" exist in CTD.Lemmas then
        Roots.add(stem+"mek")
        continueParsing =false
    end
end
end
end
end

return Roots
end

```

In the algorithm, minWL is the minimum word length to stop parsing. Turkish has words with one or two-letter words however when analyzing for two-letter words, unreliable derived relations with root lemmas are increased rapidly. The lemma sampler prunes letters from end of a derived lemma one by one to find an existing root lemma in CTD data according to the algorithm above. After the first while loop if Roots list still empty, the word is presumed ends with inflectional suffixes. And the second while loop applies parsing with checking for vowel reduction and consonant softening according to the lemma's *inflection* property. And in the algorithm, "+mek" and "+mak" suffixes are used to form dictionary lemma entry for a verb if it is found as a root lemma.

This trivial method is not sufficient to find all derived lemmas because definitions occasionally does not contain the root lemma for the derived lemma. This problem is a lexicographic deficiency to be solved by linguistics. However, in this study, if roots list is returned with a non-empty set, the root candidates added to the semantic network with unreliable derived relations for future analysis.

In this study, *antonym* relations are analyzed with sense to sense level with two different methods. *Antonym* relations are complex relationships that can vary by context. However, basic *antonym* relations connect an adjective or verb pair as antonyms (Karaağaç 2013). For example, “*clean-dirty*” is an adjective antonym pair while “*buy-sell*” is a verb antonym pair. In CTD, antonym relations are explicitly stated for the adjective pairs in the definitions with “[*sifat*] *karşıtı*” (*opposite of [adjective]*) pattern. And this pattern generally exists between the synonyms defined in the definition. When a definition of lemma, S_1 , contains the antonym pattern, the antonym is extracted and searched for its definitions to find the correct antonym sense, S_2 . If one of the senses of antonym references back to the lemma with sense S_1 , then S_1 and S_2 are antonyms.

The second method used to find antonyms is based on the *Presence* and *Absence* relations explained in previous sections. These relations are the adjective forms of a noun representing the state of existence of the noun. *Presence* relations has a pattern “[*noun*] *olan*” and *Absence* relations has a pattern “[*noun*] *olmayan*” and these patterns are compared if they have identical nouns. As an example, “*deliksiz*” (*holeless*) and “*gözlü*” (*eyed*), lemma “*eye*” is a synonym of lemma “*hole*” in Turkish, lemmas have “*deliği olmayan*” and “*deliği olan*” senses that can be connected with an *Antonym* relation. Both methods used to detect antonym relations are based on sense to sense connections without an ambiguity.

3.2. Creating the Lemma-Sense Network

In the design of semantic networks, data sources such as corpora, dictionaries and encyclopedias can be used to define the relations. In these kinds of

studies, the “linking between lemmas” approach is usually used. But this approach is insufficient to eliminate ambiguity in natural language processing since it remains at a very superficial level in terms of semantics. In a traditional dictionary, lemmas can have more than one sense, and lemmas can be related to different lemmas with the definition of its every sense. For example, lemma X may be related to lemma Y with its first sense, and lemma Z with its second sense. However, in some earlier studies determining synonymy relations from the dictionary, definitions of the senses are processed to only establish a connection directly between the lemmas, regardless of whether the lemmas are polysemous, homonymous, or having the same matching POS tags (Yazıcı and Amasyalı 2011). Synsets determined with this approach cause serious representation problems, and these problems cannot be solved later. On the other hand, in semantic networks, it is seen that better results are obtained in word sense disambiguation (WSD) when not only the lemmas but their senses are also used as a node (Anaya-Sánchez, Pons-Porrata, and Berlanga-Llavori 2007; Johansson and Nieto Piña 2015; Nieto Piña and Johansson 2016; Camacho-Collados and Pilehvar 2018; Dubossarsky, Grossman, and Weinshall 2018).

The Lemma-Sense architecture used in this study is inspired by the study of Veronis and Ide (Veronis and Ide 1990). Veronis and Ide (Veronis and Ide 1990) present a semantically more precise network design with a bipartite graph using the lemmas and their senses. Bipartite graph is a graph that nodes of the graph can be divided in two subsets of nodes that all the nodes can be connected to the nodes of the opposite subset of nodes (Buckley and Harary, 1990). As shown in Figure 3.8, bipartite graph model splits the senses of the lemma and their related lemmas in definitions which is the key concept to manage sense disambiguation. L_a indicates node of *Lemma a* while S^a_i indicates first *Sense* node of the *Lemma a* in the Figure 3.8.

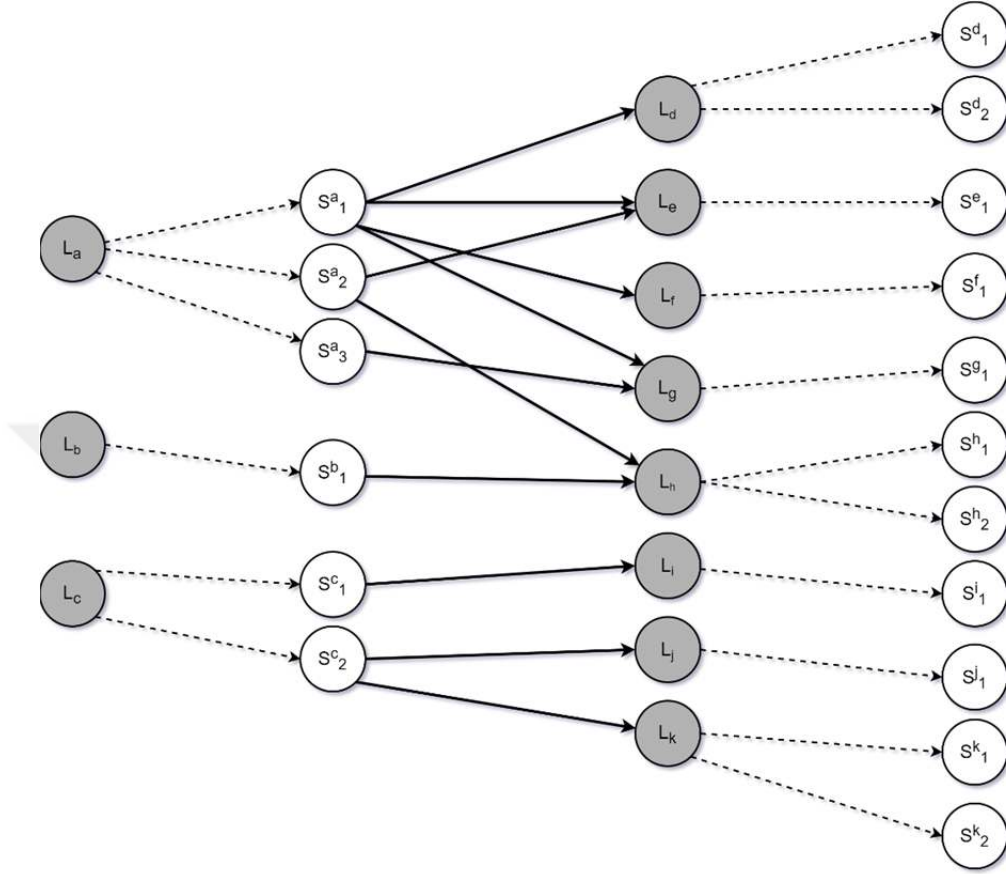


Figure 3.8. Bipartite graph model for semantic network

In this semantic network model is used in this study for the detection of weighted synonyms pairs. Dictionary lemmas and definitions are transformed into a semantic network through text processing.

There are two APIs used in text processing while creating the semantic network. Apache Commons Lang API is used for string operations to parse the definitions to lemmas. And Zemberek NLP (Akin, 2020) API used for morphologic analysis for lemmas. Neo4j graph database used for storing graph model and analyzing semantic network with graph algorithms. Advantages of using Neo4j can be listed as fast graph model creation and loading, easy to use graph algorithms on bipartite graph, and native graphical user interface (Miller 2013; Holzschuher and

Peinl 2013). Graph-based queries are applied with its Cypher query language both on native Java application, and web-based user interface.

In this network, Lemma and Sense are represented by different nodes, and a Lemma is one-way connected to its senses with the "Mean" relationship. The sense nodes establish a one-way connection with the "Mention" relationship to the Lemma that exists in the senses' explanation sentences. Since the Lemma-Sense network is created in bipartite graph architecture, neither Lemma nodes nor Sense nodes can be adjacent among themselves. An example graph representing the architecture is shown in Figure 3.9.

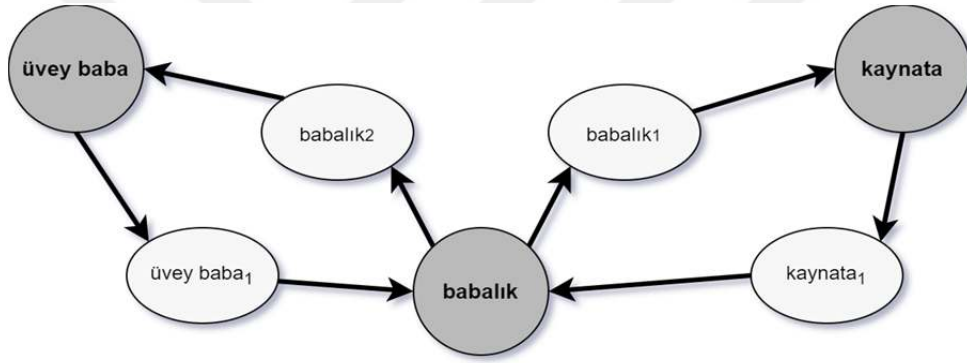


Figure 3.9. A subgraph with three lemmas based on Lemma-Sense architecture

In Figure 3.9, solid circles represent lemmas, while the hollow circles are the senses of these lemmas. The arrows, formally the relations, represent "Mean" relation if it is from a lemma to its sense, or a "Mention" relation if it is from a sense to a lemma mentioned in that sense's definition. Unlike other semantic relations, the synonym relations are symmetric where both sides of the relation must reference each other (Schmidt 2011). In Figure 3.9, the first sense of "babalık" and the first sense of "kaynata" are synonyms while the second sense of "babalık" and the first sense of "üvey baba" are synonyms. Therefore, the first and second senses of lemma "babalık" are different concepts. As a requirement of the traditional dictionary definition, the "kaynata" and "üvey baba" lemmas form

regular synonym relationships by referencing back to the “babalık” lemma that shows them again.

Each lemma (L_i) has a “Mean” relation with every sense (S_j) it has, and each S_j has a “Mention” relation with each lemma L_k that exist in its definition (if L_k and S_j have the same POS tag). Although technically not possible for the Mean relationship, Mention relationships often connect nodes with different POS tags. In order to evaluate these POS differences, the approach (Segalowitz and Lane 2000) dividing POS groups into two as content (noun, adjective, adverb, and verb) and function (conjunction, preposition, exclamation, and pronoun) can be used. According to this approach, a word belonging to the content group can be linked with the Mention relation to words belonging to the content group, but a word belonging to the function group can create a Mention relation to a lemma with only the same POS tag. In other words, different POS tags in the function group or different POS groups cannot be connected with Mention relation. In content words, a distance factor is used to distinguish the different POS tags. If a Mention relation is linked between two same POS tags (such as noun-to-noun or verb-to-verb), the distance factor value is considered to be 1, otherwise, a value greater than 1 is used. The purpose of using the distance factor is to associate the semantic distance with the POS tag and to punish the connection between words of different POS tags.

When preparing a dictionary, a lemma is expected to be explained with simpler words than itself. Following this approach, in Contemporary Turkish Dictionary (CTD), the definitions of almost all MWE lemmas are explained with simple to understand and non-compound words. When looking at definitions, MWEs are given only as synonyms, and never used in the explanation sentences of the senses. Explanations of the derived words are usually arranged using the stem form of the lemma. For example, while the lemma “göz” (*eye*) is found 304 times in definitions, the word “gözlükçü” (*optician*) is mentioned once while describing the lemma of “gözlükçülük” (*opticianry*). As a result, as the words get a suffix, the probability of finding them in the explanation sentences in the dictionary decreases.

Because of this situation, while creating the semantic network, some lemmas such as “gözlükçülük” cannot get *Mention* connection from any other *Sense* node. Similarly, we cannot expect loanwords receive a *Mention* connection except a synonym word.

Since sense nodes cannot directly point to each other in the *Lemma-Sense* architecture, determining which sense node of the lemma is affected by a synonymous relation requires a different approach. In the study, for a synonym relation, if lemma *Y* is referencing lemma *X* as a synonym in one of its senses (Y_I), but none of the senses of lemma *X* is referencing back lemma *Y*, it makes that all senses of lemma *X* are a candidate to be a synonym of sense Y_I . The right synonym sense among these candidates can be linked-to sense Y_I by finding the shortest path from candidate senses of lemma *X* to the sense Y_I on the Lemma-Sense network (Kenett et al. 2017). If two different lemmas' senses directly refer to each other, it can be said that there is a regular (bidirectional) synonym relation between these senses. However, when the dictionary definitions are not prepared with the fundamental definition rules, causing many non-regular and semi-regular synonym relations, and thus difficulties in identifying correct sense candidates for a synonym (sense matching ambiguity). Especially when it comes to metaphorical senses, it is noteworthy that the synonym relations are generally semi-regular.

Compound and MWE lemmas exist in a dictionary are also considered forming a competent semantic network between lemmas in morpho-semantic perspective. In this study, after adding lemmas and senses to the graph model, *Compound* and *Phrase* relations between the lemmas and their compound and MWE lemmas added to the graph model.

CTD dictionary data explicitly contains a list of compound words for each lemma. However, each compound lemma also exist as a lemma entry in CTD dictionary with a compounds list. And compound list of a compound lemma is a inherited compound list of the root lemma which already contains the compound lemma. A general compound set is defined and each unique compound exists in a

lemma's compound list added to the compound set. After adding lemma and sense nodes to the semantic network, every lemma node with a non-empty compound list is checked if the lemma exist in general compound set. And if a lemma with a compound list does not exist in the general compound set, then the *Compound* relations are linked between the lemma and its compound lemmas. And connected compound lemmas also are updated with a new node label "*Compound*" and their structure property in the lemma properties updated with "Compound" tag.

Compounds in CTD dictionary exist in both single-word lemma or MWE forms. As an example, lemma "ev" (*house*) has several compounds in both forms such as "yayinevi" (*publishing house*) and "ev halkı" (*household*). And beside of the compounds in MWE form, there are other MWEs defined as idioms, proverbs and compound verbs in CTD dictionary. These remaining MWEs added to the semantic network with the *Phrase* relation and they are updated with a new node label "*Phrase*" in the semantic network. However, their *structure* property in lemma properties are updated with "*Compound*" tag.

3.3. Labeling of Synonym Pairs with Confidence Indexing

Synonym relations are analyzed in sense to sense level to avoid ambiguity between lemmas' senses. First, Dijkstra's shortest path first graph algorithm used to detect synonyms are defined. Then labeling of synonym pairs with confidence indexing is explained. And the Mention distance method is explained to find shortest distance between synonym pair candidates. Mention distance method is used to find the correct synonym sense for ambiguous synonym pairs.

Dijkstra's shortest path first (SPF) algorithm is an efficient algorithm to solve the shortest path between any node S and node T on graph G which has non-negative weighted edges. SPF algorithm can be applied on directed or undirected graphs. It is closely related with Breadth First Search (BFS) and Prim's Algorithm

but with a difference. Distance to a node begins with temporary value and updated with each time a new shorter path found.

Inputs for the algorithm are node list V on graph G and links between these nodes as an adjacency list $N(v)$ with weights w (if G is directed) and finally, starting node S and ending node F .

The only output of the algorithm is the distance from starting node S to ending node F , $d(F)$. There are initial parameters to be set up before the algorithm apply on graph G . Distance between the starting node S and itself is zero, $d(S) = 0$. Distance between the starting node S and rest of the nodes V_n in the graph G is infinity. And a node list $T(v)$ is defined with an empty set for visited nodes for the algorithm. After setting up these initial parameters for SPT algorithm, pseudo-code of the process is as follows(Buckey and Harary 1990):

ALGORITHM 2: Dijkstra's SPT Algorithm

```

process  $SPT(V, N(v), S, F)$ 
   $u = S$ 
  while  $u \neq F$  do
    for each  $v$  in  $N(u)$  do
      if  $v \in V$  and  $d(v) > d(u) + w(uv)$  then
         $d(v) = d(u) + w(uv)$ 
       $V = V - \{u\}$ 
       $T = T + \{u\}$ 
      let  $u$  be node in  $V$  for which  $d(u)$  is minimum
    end
  output  $d(F)$ 
end

```

According to the pseudo-code, first u node is the starting node S . In while loop, each node in set V is processed according to minimum distance order until set V is empty.

In for loop, distance of each neighbor of node u , $d(v)$, is compared with the new distance value. If the new distance value, sum of the distance of node u and the weight of the edge between node u and node v , is smaller than $d(v)$, it is updated with new distance value.

After processing all the neighbors of the node u , node u will be removed from set V , and added to the set T . And the next u value will be node with the minimum distance in set V . When all nodes except F are processed in while loop, the distance between node S and node F can be obtained with $d(F)$.

Some evaluations should be made on the bipartite graph obtained by passing the definition sentences through text processing. For example, suppose that the lemma Y is a synonym in the description of sense X_i of lemma X .

In the semantic network, it is a serious problem to determine which of the senses of lemma Y link a synonym relation from the sense X_i . For solving this problem, it is checked whether there is a direct reference from each Sense Y_j to the sense X_i , in other words, whether it is regular.

Each synonym relation gets a confidence index value based on the structure of its definition. Input for the Algorithm 4 is the sense list of the semantic network. The conditions that can be encountered while analyzing the senses of the lemma Y are processed with the following pseudo-code:

ALGORITHM 4: Synonym Confidence Indexing Algorithm

```

for each sense  $X_i$  is properly synonyms to  $Y$  do
  if  $Y$  has no homonym and  $Y$  has only one sense ( $Y_l$ ) then
    if  $Y_l$  mentions on  $X$  then
      create synonym( $X_i$ ,  $Y_l$ ,  $I$ )
  
```

```

Else
    create synonym( $X_i$ ,  $Y_l$ , 3)
End
else if  $Y$  has at least one homonym or  $Y$  has at least two senses
    if only one  $Y_j$  mentions on  $X$  then
        create synonym( $X_i$ ,  $Y_j$ , 2)
    else if no  $Y_j$  mention on  $X$ 
        create synonym( $X_i$ ,  $Y_j$ , 4) for all  $j$  in which  $Y_j$  belongs to  $Y$ 
    else if at least two  $Y_j$  mention on  $X$ 
        create synonym( $X_i$ ,  $Y_j$ , 5) for all  $j$  in which  $Y_j$  mentions on  $X$ 
    end
end
end

```

For ease of representation, the relationships marked as confidence index N are called " ciN " for short ($N = 0, 1, 2, 3, 4, 5$). This process is applied for each sense's definition of all lemmas in the dictionary. In the algorithm, only the $ci3$ and $ci4$ relations are unidirectional. However, $ci3$ relations are not ambiguous because there is only one synonym candidate. This natural synonym is considered to have a hidden return relation. In this study, this hidden return relation is represented by $ci0$, and thus, this relationship is made regular. On the other hand, $ci4$ relations are approved as a regular synonym relation by measuring the *Mention* distance. If one or more return paths can be found between (X_i, Y_j) sense pairs, a $ci0$ relation is added between the sense pair with the smallest *Mention* distance (X_i, Y_j) . This process can be called synonym sense disambiguation. After this process, the found

relation pairs (including *ci0* and *ci4*) are considered as undirected relation labeled with 0-4 instead of *ci0-ci4*.

For the measurement of the Mention distance between lemmas (such as L_i and L_j), a path P is determined by applying the shortest path algorithm. The sum of the *Mention* distance values of all lemmas (L_k) on path P is defined as the *Mention* distance value between L_i and L_j . *Mention* distance calculation is represented mathematically by the following equation:

$$D_{Mention}(L_i, L_j) = \sum_{\forall L_k \in P(L_i, L_j)} df(L_k) * inMentions(L_k) / outMeans(L_k) \quad (3.3)$$

In Equation 3.3, the expression $inMention(L_k)$ indicates the number of *Mention* relations coming to the L_k node, $df(L_k)$ defines the distance factor of the L_k node, and the term $outMeans(L_k)$ indicates the number of sense nodes of the L_k lemma node. The path P passes on the lemma and sense nodes, respectively, on a bipartite graph-based network. Senses with different POS tags can be found in the same lemma. In this case, the $df(L_k)$ value is found by checking the POS tag of the senses linked with L_k in the P path. The calculated Mention distance is used only for synonym sense disambiguation in 0-4 relation pairs.

In the study, a different approach eliminates sense matching ambiguities in *ci5* relations. This approach is explained with the example given in Figure 3.10.

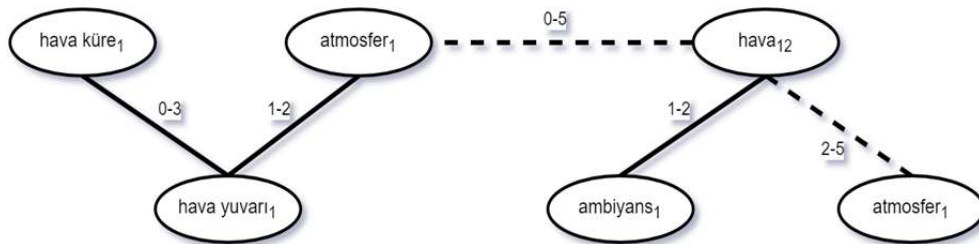


Figure 3.10. A synonym sense ambiguity of multiple *ci5* relations

The subgraph of senses in Figure 3.10 is extracted from the semantic graph as a synset when this synset analyzed manually, there are two synsets conceptually different from each other which are linked together with a 0-5 pair. The *hava*₁₂ (*air*) and the *atmosfer*₃ (*atmosphere*) senses point each other as a synonym. But since the *atmosfer*₂ (*atmosphere*) sense also has lemma “*hava*” in its definition with an improper pattern, it creates ambiguity. This situation makes two synonym relations between “*hava*” and “*atmosfer*” to be indexed with *ci5* because of multiple referencing to the same lemma's senses. There is a simple way to overcome this ambiguity, if there are two senses that reference each other with proper synonym pattern in their definitions, an ambiguity caused by improper *Mention* relations can be resolved by ignoring it. In this situation, synonym relations between *hava*₁₂ and *atmosfer*₃ are proper relations while sense *atmosfer*₂ does not have a proper reference to *hava* which can be ignored to promote the relationship between *hava*₁₂ and *atmosfer*₂. When there are no multiple references to *hava* from *atmosfer*, synonym relations between *hava*₁₂ and *atmosfer*₂ index value changed to 2-2 to raise the confidence level of the synonym relations as seen in the Figure 3.8.

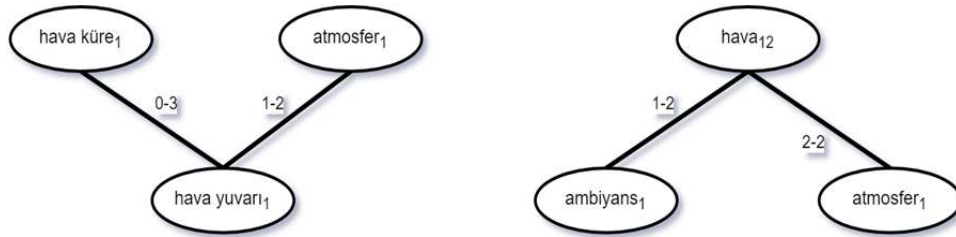


Figure 3.11. Two synsets after disambiguation of ci5 relations

All sense pairs that have a relation (X_i to Y_j) on the network, go through a bidirectional relation check. If there is no direct return, the *Mention* relation is searched. If a path is found, a new *ci0* relation (Y_j to X_i) is added to make it a pair relation. After this step, the bidirectional relation pairs between senses are

transformed into undirected ones, and these relations are labeled with a pair of confidence indices such as *1-2*, *0-4*, or *5-5*. Synonym relations indexed with *1-1*, *1-2*, *2-2* are defined as regular synonym relations because of bidirectional references with proper synonym pattern. Regular synonym relations represent that the relevant definition is prepared under the fundamental definition rules and can be interpreted as machine-readable.

Undirected synonym relations labeled with *0-1*, *0-2*, *0-3* are called semi-regular synonym relations and indicate that they can be machine-readable with a simple reference back arrangement. The most reliable approach is to find synonym relations through proper synonym patterns defined in the dictionary. In non-machine-readable dictionaries, it can be observed that synonymous couples refer back to each other outside proper patterns (improper referencing). In this study, improper referencing between two senses are tagged with *0-1* and *0-2*, while the lack of re-referencing between two synonym senses, usually the metaphorical senses, are tagged with *0-3*.

Finally, undirected relations labeled with *0-5*, *1-5*, *2-5*, *5-5*, and *0-4*, which may contain serious definition problems and ambiguities, are grouped as the regular-candidate synonyms. This group usually has mismatched and/or over-referenced sense pairs, and this makes them the most unreliable synsets of the semantic network. These problematic situations arise from linguists' personal decisions to prepare definitions in the dictionary (Jackson 2002). Therefore, these synonyms should be taken seriously and even redesigned by lexicographers to repair the semantic integrity of the dictionary.

Synonyms labeled with *0-4* are found by analyzing only unidirectional pairs. These pairs are identified by the Mention distance described above and labeled with *0-4*. However, the Mention distance cannot guarantee to find the correct sense on the network. Therefore, the *0-4* tagged relations are defined in the regular-candidate group, unlike other re-referencing problems (*0-1*, *0-2* and *0-3*).

Other regular-candidate relationships (0-5, 1-5, 2-5, and 5-5) have more serious problems. In addition to the sense matching ambiguity, in *ci5* relations, there are also some problems that need to be corrected in terms of sense granularity. This problem can arise with two different approaches: aggregation of many semantic concepts in a single definition or division of a single semantic concept into multiple definitions. The assessment of these situations with confidence indexing is discussed in detail in the next section.

3.4. Evaluation of Confidence Indexed Pairs

The sense matching problems arising from the polysemy approach of the linguists who prepared the dictionary and the suggestions for solutions to these problems are explained in this section with examples. First, semi-regular synonym pairs are analyzed. While there is a unidirectional reliable synonym relationship for the 0-1 and 0-2 pairs in the analysis, a *Mention* relation is detected improperly referenced for return. Table 3.18 shows a synonym pair example for 0-2.

Table 3.18. A synonym pair example for 0-2

Lemma	Definition
ölçü	6. Değer, itibar
değer	1. Bir şeyin önemini belirlemeye yarayan soyut ölçü, bir şeyin değdiği karşılık, kıymet

According to CTD definitions, lemma “ölçü” (*measure*) has 9 senses and lemma “değer” (*value*) has 7 senses. Only the relevant ones of these definitions are presented with their sense number in Table 1. The sixth sense of “ölçü” refers to “değer” while “değer” references back to “ölçü” in its first sense. Unlike the proper synonym pattern applied in CTD definitions, synonyms can be found at the beginning or in the middle of the sentence. And while *ölçü*₆ has the proper synonym pattern, *değer*₁ makes an improper back referencing.

On the other hand, unidirectional reference is only available in synonym pairs labeled with 0-3. They are grouped in semi-regular because it has only one candidate although there is a missing reference problem. Table 3.19 shows a synonym pair example for 0-3.

Table 3.19. A synonym pair example for 0-3

Lemma	Definition
hatırlamak	1. Anımsamak
hatırına gelmek	1. Hatırlamak , aklına gelmek

Lemma “*hatırlamak*” (*to remember*) and lemma “*hatırına gelmek*” (*to come to one's mind*) shown in Table 3.19 have only one sense. In this sample, “*hatırlamak*” and “*hatırına gelmek*” which have the same root word “*hatır*” are loan words from Arabic, while “*anımsamak*” is a Turkish lemma.

The sense of “*hatırına gelmek*” refers “*hatırlamak*” with the proper synonym pattern in its definition, while the sense of “*hatırlamak*” just references another lemma, “*anımsamak*” (*to recall*), as a synonym. If the lexicographers want to prepare a machine-readable dictionary, they should design proper references for the synonyms by resolving the improper referencing (in 0-1 and 0-2) and the lack of re-referencing (in 0-3).

It is stated above that semi-regular synonyms can be interpreted as regular synonym pairs. However, for the synonym relations having ambiguity on sense level, this assumption is not valid. These regular-candidate relations should be handled carefully, as they contain ambiguities due to ill-defined senses that involve various problems. As an example of this situation, two definitions are given in Table 3.20.

Table 3.20. A synonym pair example for 0-4

Lemma	Definition
bilezik	5. (slang) Kelepçe
kelepçe	1. Tutukluların kaçmasını önlemek için bileklerine takılan, bir zincirle tutturulmuş demir halka 2. Kablo, boru vb. şeyleri bir yere bağlı tutmak için kullanılan halka veya kelepçe

As shown in Table 3.20, lemma “*kelepçe*” (*handcuffs*) which is used as a slang meaning in the fifth sense of lemma “*bilezik*” (*bracelet*) is directly defined as a synonym. Semantically, this synonym relation should match with the first sense of “*kelepçe*” (*handcuffs*). However, a sense matching ambiguity occurs due to the lack of re-referencing in the definitions of “*kelepçe*”.

In order to solve this ambiguity, the *Mention* distance calculated for the sense candidates of “*kelepçe*” and determined that the second sense is the semantically closest to be its synonym. As it can be noticed from this example, the *Mention* distance may not solve all ambiguities in the 0-4 relations by depending on the denotations of the lemmas when it comes to slang or any metaphorical meaning.

When the X_i sense of a polysemous X lemma refers to another polysemous Y lemma, if many Y_j senses mention the X lemma with improper synonym patterns in their definitions, all the related senses are tagged with 0-5.

Naturally, by only text processing, it is not possible to predict which senses must connect. The senses of lemma “*alet*” (*tool*) and lemma “*maşa*” (*tongs*), as an example of this ambiguity, are given in Table 3.21.

Table 3.21. A synonym pair example for 0-5

Lemma	Definition
alet	4. (slang) Maşa.
maşa	3. Saçları kıvırmak, düzeltmek için elektrik veya ateşle ısıtılan maşa biçiminde alet . 4. (slang) Başkasının isteklerine, amaçlarına alet olan kimse.

As seen in Table 3.21, the fourth sense of lemma “*alet*” is calling only lemma “*maşa*” with the proper synonym pattern. On the other hand, lemma “*maşa*” is referencing lemma “*alet*” with improper patterns in its third and fourth senses. When these relations are analyzed semantically, the fourth sense of “*maşa*” is found as only synonymous with the fourth sense of “*alet*”. However, since their

definitions are written in improper synonym pattern, both of the senses of “*maşa*” are assumed as the synonym candidates of the fourth sense of “*alet*”.

In order to overcome the sense matching ambiguity, the lexicographer should prepare a definition pointing lemma “*alet*” using the proper synonym pattern for the fourth sense of “*maşa*”. While the single sense of a *Y* lemma identifies an *X* lemma with the proper synonym pattern, if at least two *X_i* senses are synonyms with the *Y* lemma, the related relationships are labeled with 1-5. The sense matching ambiguity, in this case, is shown with an example in Table 3.22.

Table 3.22. A synonym pair example for 1-5

Lemma	Definition
zarfçı	1. Tenha bir yolda yere içi doluymuş gibi görünen zarf veya cüzdan bırakan, sonra da bunları bulup alan kimseyi suçlayarak, tehdit ederek para sızdıran dolandırıcı, papelci . 2. Sokaklarda iskambil kâğıtlarıyla halkı dolandıran bir tür dolandırıcı, papelci .
papelci	1. Zarfçı .

In both definitions of lemma “*zarfçı*” (*fraud that uses an envelope to deceive people - envelopeist*) given in Table 3.22 are referring the “*papelci*” (*cardsharp*) lemma as a synonym with the proper synonym pattern. The only sense of “*papelci*”, on the other hand, directly calls “*zarfçı*” as a synonym properly. But here, a strange conflict of concepts arises.

In an old edition of the dictionary (Demiray 1969), lemma “*zarfçı*” had only one sense. In the later edition of the dictionary, adding the second sense without checking the sense matching ambiguity caused this problem. Therefore, lexicographers, when adding new semantic concepts or updating old ones, should control all the semantic conflicts that may occur.

While one of the senses of a *Y* lemma calls an *X* lemma using the proper pattern, if at least two *X_i* senses refer to the *Y* lemma, the related relationships are labeled with 2-5. The sense matching ambiguity caused by the lack of conflict control is shown in the example given in Table 3.23.

Table 3.23. A synonym pair example for 2-5

Lemma	Definition
üstelemek	1. Bir düşünce veya istek üzerinde durmak, direnmek, ısrar etmek, tekit etmek . 3. Bir isteği, bir buyruğu tekrarlamak, tekit etmek .
tekit etmek	2. Üstelemek .

As seen in Table 3.23, only the second sense of lemma “*tekit etmek*” (*to reiterate*) verb, which is a compound word by combining a loan word from Arabic and a Turkish auxiliary verb, directly refers to lemma “*üstelemek*” (*to persist*) with the proper pattern.

On the other hand, both the first and the third senses of “*üstelemek*” are referencing “*tekit etmek*” as a synonym in their definitions using the proper synonym pattern. As a result, a case similar to the previous example emerges. The only difference is that this case occurs between two polysemous lemmas. The lexicographer should consider associating the proper senses by matching them with one-to-one relations.

The most complicated example of the sense matching ambiguity occurs in 5-5 indexed pairs. While at least two senses of a *Y* lemma point an *X* lemma with the proper pattern, if at least two X_i senses call to the *Y* lemma, the related relationships are labeled with 5-5. For this case, an example is shown in Table 3.24.

As seen in Table 3.24, multiple senses of these lemmas reference each other synonymously, but it is not clear which of the senses are synonymous with each other. If the number of corresponding senses for both lemmas is the same, matching the proper senses using a unique tag may resolve the problem.

Table 3.24. A synonym pair example for 5-5

Lemma	Definition
Katman	1. (noun) Birbiri üzerinde bulunan yassıca maddelerin her biri, tabaka 2. (noun, geology) Altında veya üstünde olan kayaçlardan gözle veya fiziksel olarak az çok ayrılabilen, kalınlığı 1 santimetreden az olmayan tortul kayaç birimi, tabaka 3. (noun, sociology) Bir toplum içinde makam, şöhret, meslek vb. bakımdan ayrılan topluluklardan her biri, tabaka
Tabaka	1. (noun, geology) Katman 2. (noun) Baskı ve yazıda kullanılan, değişik boyutlarda kesilmiş kâğıt 3. (noun) Derece 4. (noun, sociology) Katman

But in Table 3.24, the number of senses to match does not overlap. On one side, all three senses of lemma “*katman*” (*layer*) refer lemma “*tabaka*” (*layer*) as a synonym, while only the first and the fourth senses of “*tabaka*” reference back to “*katman*” as a synonym. Three senses of “*katman*” can not match one-to-one with two senses of “*tabaka*”. When this situation is analyzed using the tags given in parentheses in the definitions, the second sense of *katman* matches with the first sense of *tabaka*, and also the third sense of *katman* matches with the fourth sense of *tabaka*. But none of the senses of *tabaka* is a synonym of the first sense of *katman*. Thus, it is understood that lexicographer does not use the one-to-one matching method when choosing proper senses.

After tagging with confidence indexing, the tagged bidirectional synonym relations are converted to undirected relations. On the undirected semantical network, a spanning tree-based approach is used to detect the proper synsets.

3.5. Spanning Tree-based Synset Detection

A group of synonymous words with the same POS tag has been accepted as a synset in the natural language processing literature, and the term is derived from the abbreviation of the synonym set in English (Fellbaum 1998). Grouping synonymous words within the same concept is a very important and hard task.

While a two-word synset need only one synonym relation, three synonym relationships is necessary to name a three-word synset without an ambiguity. The regular synonym relationships required by any three sense nodes (X_I - Y_I - Z_I) to be a synset are shown in Figure 3.12.

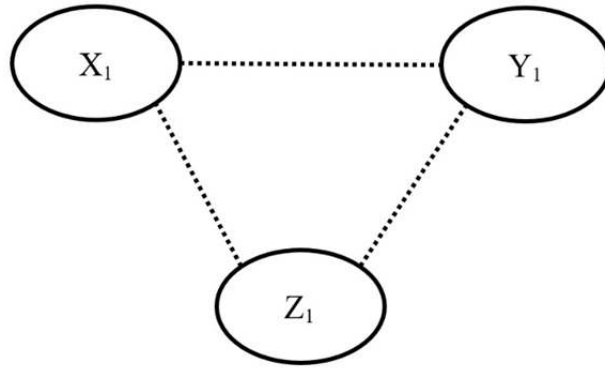


Figure 3.12. A three-word synset

As shown in Figure 3.12, to prepare a three-word synset, there should be three regular synonym relations between the three sense nodes represented in the full graph. If we generalize this case with a mathematical expression, a synset containing n -words should have regular synonym relations as much as $n * (n-1) / 2$. Based on this idea, it is possible to detect synset using synonyms pairs that are known to be regular. However, in traditional dictionaries, there may not be a sufficient number of synonym relationships for various reasons.

After analyzing the semantic network, the spanning tree algorithm is used to detect synsets. It is not a new idea for synset detection. In the study of Tarjan's (Tarjan 1972) for the first time, the connected component analysis is applied to directional graphs. Directed graphs are divided into two groups according to the connectivity: weakly connected and strongly connected. A directed graph is called "weakly connected" when there is at least two distinct nodes are not connected with in a cycle. And a directed graph G is a strongly connected graph, if every pair of

nodes in graph G exist in a cycle. Strongly connected components are the distinct strongly connected subgraphs of a graph G used in many graph application to analyze the graph structure. And SCC algorithms use depth-first search (DFS) algorithm to search a graph. Tarjan's SCC algorithm one of the algorithms that uses DFS but with a difference, it runs DFS only one time on graph G . And the algorithm makes certain that there are no merging connected components.

There are initial parameters to be set up before the algorithm apply on a graph, $G=(V,E)$. For running DFS algorithm for once, there must be a stack S for the nodes V . And the edges E must be defined for each node v as an adjacency list $N(v)$. And an integer i with initial value $i=0$ is needed to keep the index value for the nodes.

Tarjan's SCC algorithm runs in recursive approach to go depth in the node's neighbors and then collect each connected node that form a strong connected component. And the stack S keeps the list of visited nodes of v and itself to create current strongly connected component. Each node pushed in and pulled from stack S only once so the algorithm to find strongly connected components requires $O(V+E)$ time and space.

There are two indexing values for the nodes. First indexing value is the order of the node, $Num(v)$, and unique to each node but initially defined with -1 for each node. And the second indexing value $Lowlink(v)$, is the smallest indexed node in the same component as v . If v is the beginning node of a strongly connected component C of G , then $Lowlink(v)$ value equals to $Number(v)$ value because if $Lowlink(v) < Number(v)$ is true then there is at least one node that linked to v in the component C . After setting up these initial parameters for SCC algorithm starting any node v from V , pseudo-code of the procedure is as follows (Tarjan, 1972):

ALGORITHM 3: Tarjan's SCC Algorithm

```

Procedure  $SCC(v)$ 
  Lowlink( $v$ ) =  $i$ ;
  Number( $v$ ) =  $i$ ;
   $i++$ ;
   $S.push(v)$ ;
  // DFS algorithm to find SCC
  for each  $w$  in  $N(v)$  do
    if Number( $w$ ) == -1 then
       $SCC(w)$ ; // Recursive call
      Lowlink( $v$ ) =  $\min(Lowlink(v), Lowlink(w))$ ;
    else if Number( $w$ ) < Number( $v$ ) then
      if  $S.contains(w)$  then Lowlink( $v$ ) =  $\min(Lowlink(v), Numlink(w))$ ;
    end
  end
  // Collecting nodes belongs to new SCC
  if Lowlink( $v$ ) == Number( $v$ ) then
    newSCC[] = {};
    while Number( $w$ ) >= Number( $v$ ) do
      newSCC.add( $S.pull(w)$ );
    end
    output(newSCC);
  end
  // Reseting stack  $S$  and  $i$  to find next SCC
   $i=0$ ;
   $S.removeAll()$ ;
  for each  $w$  in  $V$  do
    if Number( $w$ ) == -1 then  $SCC(w)$ ;
  end
end

```

According to the pseudo-code, when a node v send to the procedure SCC, $Lowlink(v)$ and $Number(v)$ values are defined according to integer i , and i is increased for the next node in the component. Then the node v pushed in stack S . Each neighbor node w checked if the $Number(w)$ value defined. If $Number(w)$ is not defined, node w , is a new node for the component and next step will be calling procedure SCC for node w to acquire its $Number(w)$ and $Lowlink(w)$ values. And after the comparison, if $Lowlink(w)$ is smaller than $Lowlink(v)$ value, then $Lowlink(v)$ value will be updated with smaller value of $Lowlink(w)$. If $Number(w)$ value already defined for node w and node w exists in stack S , then if both $Number(w) < Number(v)$ and $Lowlink(w) < Lowlink(v)$ are true then $Lowlink(v)$ value is updated with $Lowlink(w)$. And when this DFS algorithm loop finished for neighbors of node v then a strongly connected component ready to define.

Next part of the pseudo-code, node v and its neighbors are checked if their $Number()$ and $Lowlink()$ values are equal. If for node v , $Number(v) = Lowlink(v)$ then node v is the root node of current strongly connected component of graph G . When the root node is found, a new strongly connected component is created with the nodes in the stack S that have $Number(w)$ value greater or equal to $Number(v)$. And the process outputs the new SCC.

Last part of the pseudo-code, after an SCC found, integer i is set to 0 and every remaining node removed from stack S . And procedure SCC called for another random node from graph G that is not yet numbered.

Many researchers focusing on preparing wordnet from the dictionary have followed a similar path (Mostafazadeh and Allen 2015; Ustalov, Panchenko, and Biemann 2017; Ehsani, Solak, and Yildiz 2018)(Mostafazadeh and Allen 2015; Ustalov, Panchenko, and Biemann 2017; Ehsani, Solak, and Yildiz 2018). However, the novel approach (labeling synonym relations with confidence levels, choosing some synonym relations depending on their confidence level, converting the chosen directed synonym relations into undirected ones, and then making

spanning tree-based synset detection on the undirected graph) is proposed for the first time in this study.

A representative example of the proposed approach is described with the seven lemmas given in Table 3.25.

Table 3.25. Seven lemmas and their definitions for synset detection

Lemma	Definition
Simge	1. Duyularla ifade edilemeyen bir şeyi belirten somut nesne veya işaret, alem , remiz , rumuz , timsal , sembol
Remiz*	1. Simge
Rumuz*	1. Simge
Timsal*	1. Simge
Alem*	3. Simge
Sembol*	1. Simge
Bayrak	3. (metaphor) Simge , sembol

*: loan words

When Table 3.25 is analyzed, one of the senses of the six lemmas references lemma “*simge*” (*symbol*) with the proper synonym pattern, while the definition of the only sense of “*simge*” calls back to five of the lemmas with the proper synonym pattern. Lexicographers who prepared the dictionary are designed lemma “*simge*” at the center of this synset because it is of Turkish origin. In the third sense of “*bayrak*” (*flag*), “*simge*” and “*sembol*” lemmas are synonyms with the proper synonym pattern. On the other hand, since both “*simge*” and “*sembol*” have only one sense not gone back lemma “*bayrak*”, their synonym relations are tagged with 0-2. When this example in the *Lemma-Sense* network is analyzed by the confidence indexing, its processed undirected graph structure is shown in Figure 3.13.

The set of relations given in Figure 3.13 represents only a subset of the synonym relations belonging to its regular synset. In the past, to reduce the cost of

printing of dictionaries, synonym relations in definitions were defined by the most popular words, so definitions could be created with less synonym relations. The lexicographers were assumed that the missing relationships in dictionary can be easily predicted by the reader.

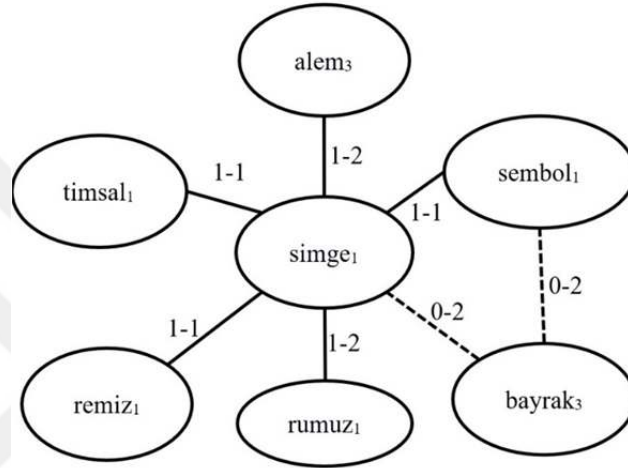


Figure 3.13. Regular synonym relations among the seven senses

For example, among the seven words in Figure 3.13, only lemma “*simge*” has a synonym relation with all other words. Considering that these seven lemmas are the elements of the same synset, only 7 synonym relations among 21 are found in the definitions. In the study, it is assumed that the sense nodes that have a spanning tree on synonym relationships form a synset. In Figure 3.8, the relationships marked with straight edges represent regular synonym relationships, while the dashed ones represent semi-regular relationships. If only regular synonym relations are used, a spanning tree is formed that resembles star topology where “*simge*” is in the center. It can be said that synsets at this level are the most reliable and can be used with almost no manual verification.

In the study, synonym relations indexed as *1-1*, *1-2*, *2-2* are defined as regular synonym relations and synsets that detected using only these relations are

labeled as "*Confidence Level 1*" (*CL-1*). As described in the previous section, semi-regular synonyms indexed with 0-1, 0-2, and 0-3 still reliable enough to form synsets even though they may require partial control because of metaphorical senses. From a cumulative perspective, synsets containing both regular and semi-regular synonym relations are marked with "*Confidence Level 2*" (*CL-2*). In the study, it is thought that regular-candidate relations labeled with 0-5, 1-5, 2-5, 5-5, 0-4 need serious manual controls because they contain fundamental problems in the definition design. Therefore, synsets containing also these relations (in addition to regular and semi-regular relations) are marked with "*Confidence Level 3*" (*CL-3*). In the example given in Figure 3.13, the synset detected at the *CL-1* level does not include the third sense of "bayrak", while the synset at the *CL-2* level contains it. Because there is no regular-candidate relation in this sample, the synset at the *CL-3* level is the same with the one at *CL-2*. From a semantic perspective, it has been determined that the synset detected at *CL-2* level can be considered as correct. However, as explained in the results and discussion, this is not always the case.

4. RESULTS AND DISCUSSIONS

In this study, a *lemma-sense* based semantic network designed with 91,363 lemma nodes connected to 121,357 nodes senses. On bipartite graph based Lemma-Sense network, 576,221 *Mention* and 121,357 *Mean* relations are established. 46,511 lemma nodes are referenced by at least one *Mention* relation, while 44,852 lemma nodes are remained disconnected. Thus, nearly half of the lemma nodes are excluded from *Mean-Mention* semantic path for shortest path analysis. All "*Sense to Sense*" relations are analyzed under this condition.

4.1. Semantic Network Analysis

Mention relations connected to a lemma node can be between zero and 18,385 (word "bir", one). Higher number of *Mention* relations connected to a lemma node is caused by two cases: lemma is a stop word or has a higher-level rank in natural language ontology. For example, lemma "bir" (*one*) is a stop word and on the semantic network it has 18,385 incoming *Mention* relations. Lemma "bir" has 13 senses with different POS tags such as noun, adjective and adverb. Although it is a stop word, it is connected to many lemmas on the semantic network with different senses. For another example, lemma "bitki" (plant) is a hypernym word for many lemmas in CTD and there are 1063 incoming *Mention* relations. Lower number of *Mention* relations connected to a lemma node indicates that lemma has a lower-level rank in natural language ontology. A lemma in absence of any incoming *Mention* relations indicates that lemma is one of a derived word, compound word, MWE or a loan word. Also, controlled defining vocabular (CDV), a small supervised set of lemmas in a language to define all lemmas in a dictionary, is the approach that triggers this condition (Xu 2012).

On compound and phrase relation linking between lemmas, 31,583 *Compound* and 15,148 *Phrase* relations created in the semantic network. These

relations are created with existing CTD data, thus, compound verbs were listed in phrase list of the lemma.

After n-gram analysis, there are 11,249 hypernym relations created in the semantic network. *Hypernym* and *Hyponym* relations are asymmetric relations (Schmidt 2011). These hyponym relations are also added to the semantic network. And 3114 *Group Of* relations are appended to the semantic network with its asymmetric *Member Of* relations.

In Table 4.1, Table 4.2 and Table 4.3, other semantic relations created with n-gram patterns are listed with a definition and total numbers according to lemma POS tags. In Table 4.1, all semantic relations are connected from a noun to a noun sense.

Table 4.1. Relations linked between nouns senses

Relationship Name	Explanation	Count
IS_A	Is A relation	11,798
NOUN_FORM_OF	Noun form of that concept	19,791
INSTANCE_OF	Instance of the object (proper noun)	886
SCIENCE_OF	Science of the object	481
MASTER_OF	Master of the object	1,118
SELLER_OF	Seller of the object	662
MAKER_OF	Maker of the object	868
MANAGER_OF	Manager of a job or task	99
OCCUPATION_OF	Occupation of that person	70
SUPPORTER_OF	Supporter of the object	26
IS_FROM	Person from that place	120
WRITTEN_WITH	Written with that language	162
MADE_WITH	Made with that matter	51

In Table 4.2, semantic relations connected between noun, adjective and adverb senses. And in Table 4.3, all semantic relations are connected between verb senses.

Table 4.2. Adjective and Adverb based semantic relations

Relationship Name	Explanation	Count
ADJECTIVE_FORM_OF	Adjective form of that concept	5,712
PRESENCE_OF	With the object	3,140
ABSENCE_OF	Without the object	2,572
RELATED_TO	Related with that object	893
AS_LIKE	As like that object	1,900
TO_BEHAVE	To behave like	304
ANTONYM	Antonym of the adjective	1,350

Table 4.3. Verb based semantic relations

Relationship Name	Explanation	Count
VERB_FORM_OF	Verb form of that concept	17,219
CAUSATIVE_OF	To be in causative form of a verb	958
OBJECT_OF	To be object of that verb	1,575
TO_BE	To be relation	6,260
TO_MAKE	To make something to become that object	1,508
QUICKLY	verb+(i)vermek auxiliary verb	1,083
ABLE_TO	Able to relation	3,963
TO_BECOME	To become that object	2182
TO_CAUSE	To cause that situation or object	249

There are 21,755 reliable derive relations and 68,735 total *Derive* relations created between root lemma and its derived lemmas. In Figure 4.1, a root lemma "göz" (*eye*) presented with its derived lemmas. All derived relations except between "gözlü" (*with an eye*) and "gözlük" (*eyeglasses*) lemmas are reliable derived relations. Derived lemma "gözlük" is connected to the longest root lemma

"gözlü" and its actual root lemma is "göz". Unreliable derive relation between lemmas "gözlü" and "gözlük" is extracted by receiving the longest root for the lemma "gözlük". And the longest root is the lemma "gözlü". The problem is occurred because of combinations in derivational suffixes.

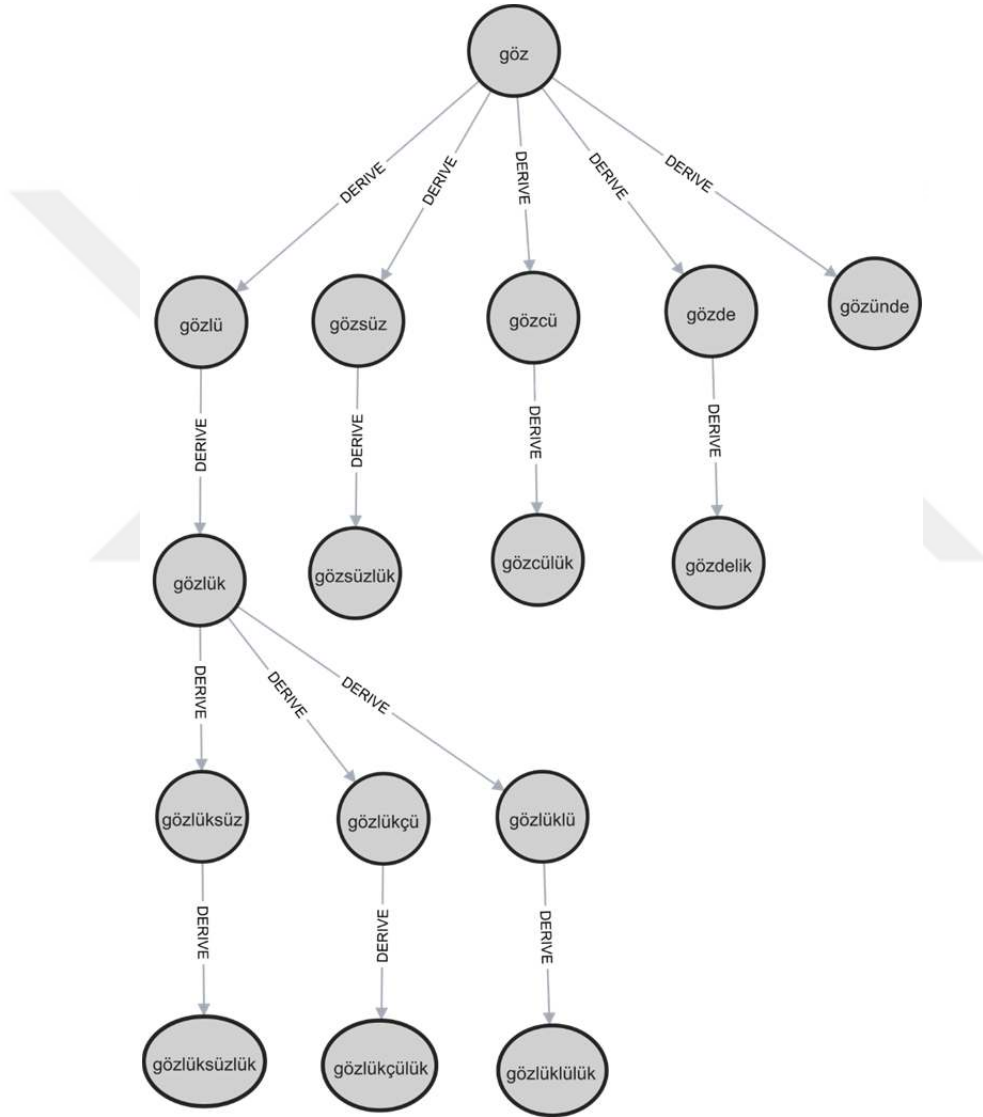


Figure 4.1. Root lemma "göz" and some of its derivative lemmas

4.2. Indexing Analysis of Synonym Relations

After text processing of the definitions, 29,375 directed synonyms are found. According to confidence indexing analysis, the distribution of the number of the found relations are given according to their confidence index pairs in Table 4.4.

Table 4.4. The distribution of found synonym relations

		Confidence Index (CI)				
		1	2	3	4	5
CI	0	436	548	9,602	4,903	168
	1	5,863	4,253	-	-	768
	2	-	2,240	-	-	435
	5	-	-	-	-	159

In Table 4.4, one-way numerical representation is preferred, such as (a, b) instead of (b, a) for $a < b$. Therefore, the diagonal bottom is left blank. While reading from the table, the small number in the relationship is searched in the row and the large number in the column. For example, the number of relations with $1-2$ indexes (which means the same thing as $2-1$) in the network is 4,253.

The first row in the Table 4.4 shows semi-regular synonyms ($0-1$, $0-2$ and $0-3$) and other relationships with no back reference. As seen in the table, the columns labeled with 3 and 4 have only relations with 0 because 3 and 4 indexed synonym relations are unidirectional. The regular synonym relations such as $1-1$, $1-2$ and $2-2$ are the most reliable synonym group. $ci5$, the least reliable group has smallest number of synonym pairs in total according to other confidence index groups. According to the table, the most common synonyms are $(0, 3)$, $(1, 1)$, $(0-4)$, $(1, 2)$, $(2, 2)$, respectively.

Based on the analysis in the study, it is determined that regular and semi-regular pairs indicate the most reliable synonyms that can be detected automatically from the dictionary. Only some of the $ci4$ relations can be reliable, but ambiguities in $ci5$ relations cannot be resolved by automated methods because

of some design errors. One error type is the matching synonym pairs with different POS tags.

In the study, while analyzing synonymous couples determined by text processing, 602 of them have POS tag mismatch. As an example of this situation, Table 4.5 shows the relationship between “*yarı*” (*half*) and “*nısıf*” (*half in Arabic*) lemmas.

Table 4.5 An example for POS tag mismatch

Lemma	Definition
yarı	1. (adj) Bir bütünü oluşturan iki eşit parçadan her biri, nısıf . 2. (adj) Bir şeyin yarısı kadar olan, yarım olan. 3. (noun) Devre arası. 4. (adv) Gereğinden az, tam olmayarak.
nısıf	1. (noun) Yarı

According to the definitions given in Table 4.5, the third sense of “*yarı*” and the single sense of the “*nısıf*” matches in POS tags, but they are not synonym to each other semantically. On the other hand, the first sense of “*yarı*” and the only sense of “*nısıf*” are synonymous in semantic manner. While this pair should be labeled with 1-2, they are not added to any synonym list in the study because of POS tag mismatch.

Some special cases encountered during synset detection process in the study are tried to be explained with examples. While few exceptions are observed at *CL-1* and *CL-2* levels that contain regular and semi-regular synonym relations, at *CL-3* level, the amount and diversity of irregularities increased. In the first example case shown in Figure 4.2, there is a synset containing two different senses belonging to the same lemma although it is at *CL-1* level.

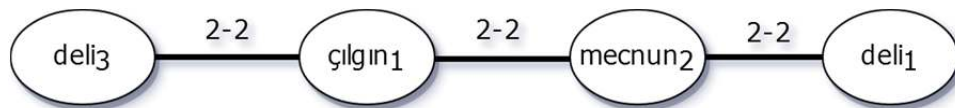


Figure 4.2. The undirected graph of first example

All synonym relations in Figure 4.2 are labeled with 2-2. It means that the synset to find is at *CL-1*, the most reliable level for synonym relations. However, the presence of two different senses of “*deli*” (*insane*) in the same synset indicates a technical error about sense definitions. When all analyses are completed, 18, 115, and 592 repeated lemma are found at *CL-1*, *CL-2*, and *CL-3* levels, respectively. According to this statistic, the number of synsets with repeated lemma increased almost 5 times with each level up. Although no other type of problem can be identified at the *CL-1* level in the study, the first example show that if the dictionary is prepared without following the fundamental definition rules, there may be some errors in each confidence level. In the second example, nine lemmas and their definitions are listed in Table 4.6.

Table 4.6. The lemmas and the synonym definitions of the second example

Lemma	Definition
doğrudan doğruya	1. dolaysız, araçsız, aracısız , araya başka bir şey girmeden, resen
elden	1. doğrudan
aracısız	2. aracı olmadan, doğrudan, direct
araçsız	2. araç olmaksızın, vasıtasız bir biçimde, bilavasita, doğrudan doğruya
bilavasita	2. birinin aracılığı olmadan, doğrudan doğruya, aracısız
doğrudan	2. aracısız olarak, herhangi bir aracı kullanmadan
resen	2. bağımsız olarak, kimseye bağlı olmaksızın
vasıtasız	2. doğrudan
direkt	3. doğrudan, doğrudan doğruya

In Table 4.6, the related words referenced to each others are marked as bold. It appears that not all definitions do reference other lemmas. The undirected graph obtained after the approach (text processing, confidence indexing, converting the graph into the undirected one) is shown in Figure 4.3.

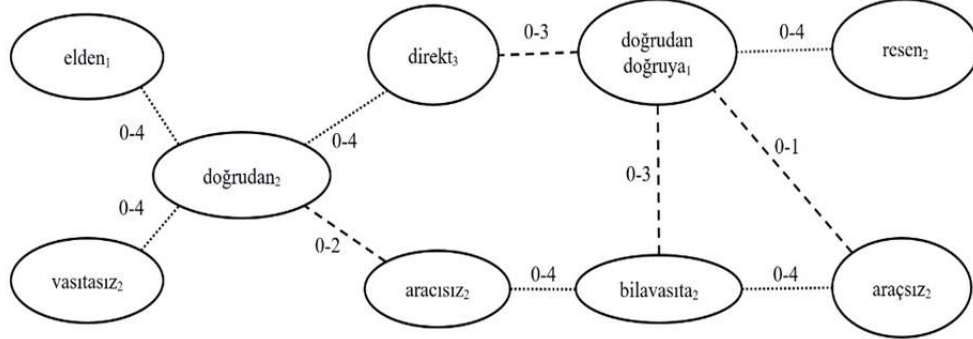


Figure 4.3. The undirected graph of second example

In Figure 4.3, although there is no regular synonym pair, the absence of any *ci5* relationship provided a reliable synset to be found at *CL-3*. If the same graph is evaluated at the *CL-2* level, all synonym relationships labeled with *0-4* are ignored. As a result, some sense nodes such as *elden₁* (*with hand*), *vasıtasız₂* (*without an intermediary*), and *resen₂* (*independently*) are considered as a synset alone. Besides, the remaining connected nodes are found in two separate synsets. Although the synsets defined in KeNet (Ehsani, Solak, and Yildiz 2018) for this example looks like ones determined at *CL-2* level in this study, semantically, the correct synset for this graph can be found at the *CL3* level. This example shows that some synsets at *CL-3* may also be completely correct.

In the study, a reliable synset detection at the *CL-3* level is required only the presence of relationships labeled with regular, semi-regular, or *0-4*. Relationships labeled with *0-4*, which have the lack of re-referencing problem, have a confidence level lower than the others. In the *Lemma-Sense* network, 12,217 definitions with the lack of re-referencing problem have been identified. Because of this ambiguity, 48,878 candidate senses are marked with *ci4*. It is determined that only 5,261 of the identified synonyms have at least one indirect return path. For the ones with more than one return path, the shortest path approach (minimum *Mention* distance) has been applied for resolving the ambiguity, and all found pairs are labeled with *0-4*. For each pair, the approach is repeated using different

distance factor values (1, 2, 3, 5, and 10). Among these 5,261 synonym pairs, 200 pairs are chosen randomly, and the samples are manually controlled whether they are matched with the correct sense. As a result of the control, the numbers of the correct matched senses by different distance factors are presented in Table 4.7.

Table 4.7. The numbers of correct matched senses by different distance factors

df	1	2	3	5	10
correct	72	75	74	81	74

As seen in Table 4.7, when the df value increases up to 5, an increase in finding the correct synonyms occurs. Therefore, the df value is used as 5 in the study. This proves that semantic relations occur intensely between lemmas with the same POS tag. Besides, it can be interpreted that the rate of ambiguity in the *ci4* relationships can be corrected by automated methods (synonym sense disambiguation) at the level of 40%, and similarly, synsets at the *CL-3* level without any *ci5* relation can be reliable with the same rate. However, this low success rate shows that the ambiguities of the *0-4* pairs are irregular to be solved by automated methods. Relations labeled *0-4*, which can be found in *CL-3* level synsets, are often incorrectly matched, and therefore two separate synsets can be mistakenly interpreted as a single synset. Finally, in the third example, the *ci5* relations are shown in Table 4.8.

Table 4.8. The lemmas and the definitions of the third example

Lemma	Definition
hedonizm	1. hazcılık
hazcılık	1. zevki, insan hayatının tek değer ve amacı sayan, haz veren her şeyin iyi olduğunu kabul eden öğretisi, hedonizm 2. Hazza, fiziksel zevke hastalık derecesinde düşkünlük, hedonizm 3. Ekonomik etkinliğin, hazzın en yüksek derecesine varacak biçimde geliştirilmesi öğretisi, hedonizm

When Table 4.8 is analyzed, three different senses of lemma “*hazcılık*” (hedonism) reference back to the only sense of “*hedonizm*” (hedonism). In order to easily see the architectural problem, the relations of the senses are shown in the graph model in Figure 4.4.

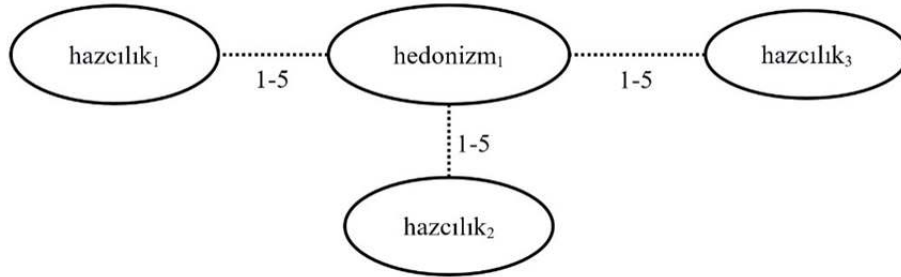


Figure 4.4. The undirected graph of the third example

The example in Figure 4.4 shows an interesting problem. If a lemma indicates three different concepts; according to the basic expectation in natural language, that lemma must have at least three senses. But in this example, “*hedonizm*” seems to have one sense. This ambiguity is observable in the graph model given in Figure 4.4. This problem can only be solved by the linguist doing a modeling that depends on the synset architecture. But if the linguist thinks that this ambiguity is easily solved by people, solving this problem is not possible. Although semantic updates in the dictionary need linguists, in order to have a flawless dictionary, the effects of the updates on the mathematical model of the dictionary should be followed with computational systems.

4.3. Comparison of the Results

The synsets detected in the study are evaluated in confidence index perspective by comparing with KeNet. KeNet is the only open access Turkish WordNet study in the literature and it uses the same dictionary resource CTD. In the comparison, synsets only containing at least two lemmas are considered. A common vocabulary is determined, and all synsets containing at least one lemma

that is not in that vocabulary are excluded from the comparison. The comparison focuses on to find the match, the superset and the subset values between the synsets detected at three different levels (*CL-1*, *CL-2*, *CL-3*) in this study and KeNet's synsets. Accordingly, the results of the comparison are shown in Table 4.9.

Table 4.9. Comparison of two studies for different confidence levels

Level	Total	Superset	Match	Subset
CL-1	8,772	346	4,770 (54.38%)	1,850
CL-2	10,969	964	6,321 (57.63%)	1,296
CL-3	10,766	1,031	6,314 (58.65%)	1,043

In Table 4.9, “*Total*” shows the total number of synsets detected based on confidence levels in this study, “*Superset*” is the number of synsets where each one covers at least one synset of KeNet, “*Match*” represents the number of the synsets that match in both studies and its ratio to total, and “*Subset*” indicates the number of synsets in KeNet in which each one covers at least one synset detected in this study. In terms of match ratio although the most successful line of Table 4.9 seems to be *CL-3* when the number of changes in total and superset numbers are evaluated, it can be said that the most serious match with KeNet is at the level of *CL-2*. In *CL-3* level, the increase in the supersets while the total value decreases can be interpreted as that most synsets merge. This means that a serious problem has arisen, such as the collection of different semantic concepts under the same synset. When the numbers are analyzed, perhaps the most important finding is that the synsets of an automated WordNet can include the synsets of a WordNet organized under the supervision of experts. Based on this situation, when some of the synsets considered as the superset are visually examined, serious semantic divisions are determined at the synsets in KeNet. While *CL-3* has many giant synsets (such as 1,481 - 2,414 - 2,568 senses), the largest synset at *CL-1* includes only 32 senses. On the other hand, at the *CL-2* level, there are 22 synsets larger

than the largest synset at *CL-1*, and the largest synset contains 898 senses. Therefore, it is thought that computer-assisted services reinforced with the confidence levels presented in this study can make serious contributions to the WordNet studies.



5. CONCLUSION

There are many studies in the literature that make automatic synset detection from monolingual dictionaries. The most serious problem in these studies is that the detected synsets need an expert to validate. In this study, a special confidence index approach is proposed to minimize human labor to be spent for validation of automated detected synsets. According to the approach, synonym relations are labeled with integer values between 0 and 5, then the detected synsets with three types of confidence levels, such as *CL-1*, *CL-2* or *CL-3*. It can be said that there is no need for manual validation (except for repeated use of sense) for *CL-1* that contain the most regular synonym relations. It has been observed that the synsets labeled with *CL-2* often do not require manual validation like *CL-1* marked synsets, but they may contain serious back referencing and rarely sense granularity problems. On the other hand, *CL-3* synsets may contain various problems. In synsets labeled with *CL-3*, the linguist can understand the source of the problem and try to solve it especially by looking at the relation label. The most challenging dictionary problems can be listed as lack of re-referencing, improper referencing, repeated use of sense, POS tag mismatch, sense granularity.

On the other hand, it has been determined that the dictionary definitions used as the source are seriously problematic. While there are 121,357 senses and definition sentences in the dictionary, it has been determined that there are POS tag mismatches in 602 synonym relations between them, as well as many ambiguities (1,898 *ci5* and 12,217 *ci4*) and the lack of re-referencing (9,982 *ci0*). Therefore, the perception that the dictionary contains roughly 40% of problematic definitions arises. If we interpret this ratio in reverse, it can be said that the dictionary used in the study is approximately 60% machine-readable.

The findings of the study are compared with ones of KeNet, which is the only Turkish WordNet with open access in the literature. The *CL-1* level suggests a relatively small size synsets with few senses as the most reliable level.

Nevertheless, at the *CL-2* level, there are more synsets matched with KeNet synsets. Raising from *CL-2* to *CL-3* causes a small decrease in the number of synsets that match with KeNet synsets while creating giant synsets by merging unrelated synsets. Consequently, the *CL-3* level can be used to detect erroneous definitions for lexicographers who designed the dictionary rather than automatic synset detection. As a result, assistant services designed for experts in a new WordNet preparation work can be made more successful with the support of the confidence levels presented in this study.



REFERENCES

- Akın, Ahmet A. (2013) 2020. *Zemberek: NLP Tools for Turkish*. Java. <https://github.com/ahmetaa/zemberek-nlp>.
- Alexeyevsky, Daniil, and Anastasiya V Temchenko. 2016. “WSD in Monolingual Dictionaries for Russian WordNet.” In *Proceedings of the 8th Global WordNet Conference*, 6.
- Amasyalı, Mehmet Fatih. 2005. “Automatic Construction of Turkish WordNet.” In *Proceedings of the IEEE 13th*, 248–251. Signal Processing and Communications Applications Conference.
- Anaya-Sánchez, Henry, Aurora Pons-Porrata, and Rafael Berlanga-Llavori. 2007. “TKB-UO: Using Sense Clustering for WSD.” In *Proceedings of the 4th International Workshop on Semantic Evaluations - SemEval '07*, 322–25. Prague, Czech Republic: Association for Computational Linguistics. <https://doi.org/10.3115/1621474.1621544>.
- Bilgin, Orhan, Ozlem Cetinoglu, and Kemal Oflazer. 2004. “Building a Wordnet for Turkish.” *Romanian Journal of Information Science and Technology* 7 (1–2): 163–172.
- Bond, Francis, and Ryan Foster. 2013. “Linking and Extending an Open Multilingual Wordnet.” In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 1352–62.
- Bosch, Sonja E., and Marissa Griesel. 2017. “Strategies for Building Wordnets for Under-Resourced Languages: The Case of African Languages.” *Literator* 38 (1). <https://doi.org/10.4102/lit.v38i1.1351>.
- Buckey, Fred, and Frank Harary. 1990. *Distance in Graphs*. Addison-Wesley Publishing Company.
- Budanitsky, Alexander, and Graeme Hirst. 2006. “Evaluating Wordnet-Based Measures of Lexical Semantic Relatedness.” *Computational Linguistics* 32 (1): 13–47.

- Camacho-Collados, Jose, and Mohammad Taher Pilehvar. 2018. "From Word to Sense Embeddings: A Survey on Vector Representations of Meaning." *ArXiv:1805.04032 [Cs]*, May. <http://arxiv.org/abs/1805.04032>.
- Chodorow, Martin S., Roy J. Byrd, and George H. Heidorn. 1985. "Extracting Semantic Hierarchies from a Large On-Line Dictionary." In *Proceedings of the 23rd Annual Meeting on Association for Computational Linguistics*, 299–304. Association for Computational Linguistics.
- Collins, Allan M., and Elizabeth F. Loftus. 1975. "A Spreading Activation Theory of Semantic Processing." *Psychological Review* 82: 407–28.
- Comrie, Bernard. 1989. *Language Universals and Linguistic Typology: Syntax and Morphology*. 2nd ed. University of Chicago Press. <https://books.google.com.tr/books?id=kf-shTfcaPEC>.
- Çotuksöken, Yusuf. 2011. *Yapı ve İşlevlerine Göre Türkiye Türkçesi'nin Ekleri*. 1st ed. Papatya Yayıncılık.
- Demiray, Kemal. 1969. *Türkçe Sözlük*. 5th ed. 293. Ankara: Türk Dil Kurumu Yayınları.
- Dubossarsky, Haim, Eitan Grossman, and Daphna Weinshall. 2018. "Coming to Your Senses: On Controls and Evaluation Sets in Polysemy Research." In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1732–1740.
- Ehsani, Razieh, Ercan Solak, and Olcay Taner Yildiz. 2018. "Constructing a WordNet for Turkish Using Manual and Automatic Annotation." *ACM Transactions on Asian and Low-Resource Language Information Processing* 17 (3): 1–15. <https://doi.org/10.1145/3185664>.
- Ercan, Gonenc, and Farid Haziye. 2019. "Synset Expansion on Translation Graph for Automatic Wordnet Construction." *Information Processing & Management* 56 (1): 130–50. <https://doi.org/10.1016/j.ipm.2018.10.002>.
- Fellbaum, Christiane. 1998. *WordNet: An Electronic Lexical Database*. Cambridge: MIT Press.

- Finlayson, Mark Alan. 2014. "Java Libraries for Accessing the Princeton Wordnet: Comparison and Evaluation." In *Proceedings of the Seventh Global Wordnet Conference*, 78–85.
- Gonalo Oliveira, Hugo, and Paulo Gomes. 2014. "ECO and Onto.PT: A Flexible Approach for Creating a Portuguese Wordnet Automatically." *Language Resources and Evaluation* 48 (2): 373–93. <https://doi.org/10.1007/s10579-013-9249-9>.
- Güngör, Onur, and Tunga Güngör. 2007. "Türke İin Bilgisayarla İřlenebilir Sözlük Kullanarak Kavramlar Arasındaki Anlamsal İliřkilerin Belirlenmesi." *Akademik Biliřim Konferansı* 1 (1): 1–13.
- Hamp, Birgit, and Helmut Feldweg. 1997. "GermaNet - a Lexical-Semantic Net for German." In *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, 9–15.
- Holzschuher, Florian, and René Peinl. 2013. "Performance of Graph Query Languages: Comparison of Cypher, Gremlin and Native Access in Neo4j." In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, 195–204. ACM. <http://dl.acm.org/citation.cfm?id=2457351>.
- Horak, Aleřs, Karel Pala, Adam Rambousek, and Martin Povolny. 2006. "DEBVisDic – First Version of New Client-Server Wordnet Browsing and Editing Tool." In *Proceedings of the Third International WordNet Conference*, 325–28.
- Jackson, Howard. 2002. *Lexicography: An Introduction*. London ; New York: Routledge.
- Johansson, Richard, and Luis Nieto Piņa. 2015. "Embedding a Semantic Network in a Word Space." In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1428–33. Denver, Colorado: Association for Computational Linguistics. <https://doi.org/10.3115/v1/N15-1164>.

- Jurafsky, Dan, and James H. Martin. 2008. *Speech and Language Processing*. 2nd ed. Pearson.
- Karaağaç, Günay. 2013. *Dil Bilimi Terimleri Sözlüğü*. 1. baskı. Türk Dil Kurumu yayınları 1066. Ankara: Türk Dil Kurumu Yayınları.
- Kashgary, Amira D. 2011. "The Paradox of Translating the Untranslatable: Equivalence vs. Non-Equivalence in Translating from Arabic into English." *Journal of King Saud University - Languages and Translation* 23 (1): 47–57. <https://doi.org/10.1016/j.jksult.2010.03.001>.
- Katz, Jerrold J., and Jerry A. Fodor. 1963. "The Structure of a Semantic Theory." *Language* 39 (2): 170. <https://doi.org/10.2307/411200>.
- Kenett, Yoed N., Effi Levi, David Anaki, and Miriam Faust. 2017. "The Semantic Distance Task: Quantifying Semantic Distance with Semantic Network Path Length." *Journal of Experimental Psychology: Learning, Memory, and Cognition* 43 (9): 1470–89. <https://doi.org/10.1037/xlm0000391>.
- Kozima, Hideki, and Teiji Furugori. 1993. "Similarity between Words Computed by Spreading Activation on an English Dictionary." In *Proceedings of the Sixth Conference on European Chapter of the Association for Computational Linguistics*, 232–239. Association for Computational Linguistics. <http://dl.acm.org/citation.cfm?id=976772>.
- Kozima, Hideki, and Akira Ito. 1997. "Context-Sensitive Word Distance by Adaptive Scaling of a Semantic Space." *AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE SERIES 4*, 111–124.
- Miller, Justin J. 2013. "Graph Database Applications and Concepts with Neo4j." In *Proceedings of the Southern Association for Information Systems Conference*. Atlanta, USA.
- Mostafazadeh, Nasrin, and James F. Allen. 2015. "Learning Semantically Rich Event Inference Rules Using Definition of Verbs." In *Computational Linguistics and Intelligent Text Processing*, edited by Alexander Gelbukh, 402–16. Lecture Notes in Computer Science. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-18111-0_30.

- Nieto Piña, Luis, and Richard Johansson. 2016. "Embedding Senses for Efficient Graph-Based Word Sense Disambiguation." In *Proceedings of TextGraphs-10: The Workshop on Graph-Based Methods for Natural Language Processing*, 1–5. San Diego, CA, USA: Association for Computational Linguistics. <https://doi.org/10.18653/v1/W16-1401>.
- Oliveira, Hugo Goncalo, Diana Santos, and Paulo Gomes. 2009. "Relations Extracted from a Portuguese Dictionary: Results and First Evaluation." In *Proceedings of 14th Portuguese Conference on Artificial Intelligence*, 13.
- Oliver, Antoni, and Salvador Climent. 2012. "Parallel Corpora for WordNet Construction: Machine Translation vs. Automatic Sense Tagging." In *Computational Linguistics and Intelligent Text Processing*, edited by Alexander Gelbukh, 7182:110–21. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-28601-8_10.
- Orhan, Zeynep, Ilknur Pehlivan, Volkan Uslan, and Pinar Önder. 2011. "Automated Extraction of Semantic Word Relations in Turkish Lexicon." *Mathematical and Computational Applications* 16 (1): 13.
- Paiva, Valeria de, and Livy Real. 2016. "An Overview of Portuguese WordNets." In *Proceedings of the 8th Global WordNet Conference*.
- Putra, Desmond Darma, Abdul Arfan, and Ruli Manurung. 2008. "Building an Indonesian WordNet." In *Proceedings of the Second International MALINDO Workshop*.
- Quillian, M. Ross. 1969. "The Teachable Language Comprehender: A Simulation Program and Theory of Language." *Communications of the ACM* 12 (8): 459–76. <https://doi.org/10.1145/363196.363214>.
- Sagot, Benoît, and Darja Fišer. 2008. "Building a Free French Wordnet from Multilingual Resources." In *Proceedings of OntoLex*, 14–19.
- Schmidt, Gunther. 2011. *Relational Mathematics*. Vol. 132. Encyclopedia of Mathematics and Its Applications. Cambridge; New York: Cambridge University Press.

- Segalowitz, Sidney J., and Korri C. Lane. 2000. "Lexical Access of Function versus Content Words." *Brain and Language* 75 (3): 376–89. <https://doi.org/10.1006/brln.2000.2361>.
- Şerbetçi, Ayşe, Zeynep Orhan, and İlknur Pehlivan. 2011. "Extraction of Semantic Word Relations in Turkish from Dictionary Definitions." In *Proceedings of the ACL 2011 Workshop on Relational Models of Semantics*, 11–18. Association for Computational Linguistics.
- Sofia, Stamou, Oflazer Kemal, Pala Karel, Christodoulakis Dimitris, Cristea Dan, Tufts Dan, Koeva Svetla, Totkov George, Dutoit Dominique, and Grigoriadou Maria. 2002. "Balkanet: A Multilingual Semantic Network for the Balkan Languages." In *Proceedings of the 1st Global WordNet Association Conference*.
- Stanchev, Lubomir. 2012. "Building Semantic Corpus from WordNet." In *Bioinformatics and Biomedicine Workshops (BIBMW), 2012 IEEE International Conference On*, 226–231. IEEE. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6470308.
- Tarjan, Robert E. 1972. "Depth-First Search and Linear Graph Algorithms." *SIAM J. Comput.* 1: 146–60.
- Thoongsup, Sareewan, Kergit Robkop, Chumpol Mokarat, Tan Sinthurahat, Thatsanee Charoenporn, Virach Sornlertlamvanich, and Hitoshi Isahara. 2009. "Thai WordNet Construction." In *Proceedings of the 7th Workshop on Asian Language Resources - ALR7*, 139–44. Suntec, Singapore: Association for Computational Linguistics. <https://doi.org/10.3115/1690299.1690319>.
- Thorat, Sushrut, and Varad Choudhari. 2016. "Implementing a Reverse Dictionary, Based on Word Definitions, Using a Node-Graph Architecture." *ArXiv Preprint ArXiv:1606.00025*. <https://arxiv.org/abs/1606.00025>.

- Ustalov, Dmitry, Alexander Panchenko, and Chris Biemann. 2017. "Watset: Automatic Induction of Synsets from a Graph of Synonyms." In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1579–1590. Vancouver, Canada: Association for Computational Linguistics. <https://doi.org/10.18653/v1/P17-1145>.
- Veronis, Jean, and Nancy M. Ide. 1990. "Word Sense Disambiguation with Very Large Neural Networks Extracted from Machine Readable Dictionaries." In *Proceedings of the 13th Conference on Computational Linguistics-Volume 2*, 389–394. Association for Computational Linguistics.
- Vossen, Piek, ed. 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Dordrecht: Springer Netherlands. <https://doi.org/10.1007/978-94-017-1491-4>.
- Widdows, Dominic, Scott Cederberg, and Beate Dorow. 2002. "Visualisation Techniques for Analysing Meaning." In *Text, Speech and Dialogue*, edited by Petr Sojka, Ivan Kopeček, and Karel Pala, 2448:107–14. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-46154-X_14.
- Widdows, Dominic, and Beate Dorow. 2002. "A Graph Model for Unsupervised Lexical Acquisition." In *Proceedings of the 19th International Conference on Computational Linguistics-Volume 1*, 1–7. Association for Computational Linguistics. <http://dl.acm.org/citation.cfm?id=1072342>.
- Widdows, Dominic, and Beate Dorow. 2005. "Automatic Extraction of Idioms Using Graph Analysis and Asymmetric Lexicosyntactic Patterns." In *Proceedings of the ACL-SIGLEX Workshop on Deep Lexical Acquisition - DeepLA '05*, 48–56. Ann Arbor, Michigan: Association for Computational Linguistics. <https://doi.org/10.3115/1631850.1631856>.
- Woods, W A, and Bolt Beranek. 1975. "What's in a Link: Foundations for Semantic Networks," 76.

- Xu, Hai. 2012. "A Critique of the Controlled Defining Vocabulary in *Longman Dictionary of Contemporary English*." *Lexikos* 22 (1). <https://doi.org/10.5788/22-1-1013>.
- Yazıcı, Emre, and Mehmet Fatih Amasyalı. 2011. "Kavramlar Arası Anlamsal İlişkilerin Türkçe Sözlük Tanımları Kullanılarak Otomatik Olarak Çıkartılması." *EMO Bilimsel Dergi* 1 (1): 1–14.
- Zhu, Huichun. 2014. "Cross-Linguistic Evidence for Cognitive Foundations of Polysemy." *Proceedings of the Annual Meeting of the Cognitive Science Society* 36: 7.
- Zülfikâr, Hamza. 2011. *Terim Sorunları ve Terim Yapma Yolları*. 2nd ed. Türk Dil Kurumu Yayınları. Ankara: Türk Dil Kurumu.

CURRICULUM VITAE

Erhan TURAN was born in ADANA, in 1985. He received a Bachelor's degree in 2009 and M.Sc. degree in 2014 from Çukurova University Department of Computer Engineering. After graduation, in 2010, he started to work at Osmaniye Korkut Ata University as a research assistant and in 2014, temporary transferred to Çukurova University as a research assistant to fulfill his Ph.D. study. During the Ph.D. study at Çukurova University, he had worked as Project Assistant for the TÜBİTAK Project with the number 215E256.