REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

CORPUS-DRIVEN SEMANTIC RELATIONS EXTRACTION FOR TURKISH LANGUAGE

TUĞBA YILDIZ

PhD. THESIS
DEPARTMENT OF COMPUTER ENGINEERING
PROGRAM OF COMPUTER ENGINEERING

ADVISER
ASSOC. PROF. DR. BANU DİRİ

İSTANBUL, 2014

REPUBLIC OF TURKEY

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

CORPUS-DRIVEN SEMANTIC RELATIONS EXTRACTION FOR TURKISH LANGUAGE

A thesis submitted by Tuğba YILDIZ partial fulfillment of the requirements for the degree of **PhD** is approved by the committee on 24.07.2014 in Department of Computer Engineering, Computer Engineering Program.

Thesis Adviser

Assoc. Prof.Dr. Banu Diri

Yıldız Technical University

Approved By the Examining Committee

Assoc. Prof. Dr.Banu Diri

Yıldız Technical University _____

Assist. Prof. Dr. Zeynep Orhan

Fatih University _____

Assist. Prof. Dr. M. Fatih Amasyalı

Yıldız Technical University _____

Prof. Dr. A. Coşkun Sönmez

İstanbul Technical University _____

Assoc. Prof. Dr.Songül Albayrak

Yıldız Technical University _____

ACKNOWLEDGEMENTS

I am deeply thankful to my advisor Assoc. Prof. Dr. Banu Diri for her tremendous support and mentorship during my thesis journey. She did not only help me during my research but encouraged me to make it better. With her guidance and knowledge, I was able to look at my research from different perspective.

I would like to thank my committee members Assist. Prof. Dr. M.Fatih Amasyalı and Assist. Prof. Dr. Zeynep Orhan for their support and contribution to my thesis. They were always available to answer my questions and encourage me this tough work.

My colleague at Bilgi University, Assist. Prof. Dr. Savaş Yıldırım, did not only help me with my project but taught me different perspectives. I thank him for his support.

My dearest friends, Sevda, Aslı and Pınar always supported my journey by sharing joy and their admiration to encourage me.

My dear family, my sister Jülide, my mummies Semra and Gül, daddy Tarık would like thank them all for their support for encouraging me and giving their full support with my decision.

During my PhD period, I was lucky to be receive the most amazing gifts of my life; my 5-year-old daughter Nehir and 1-year-old Nil. I thought it would be hard to complete my degree with two pregnancies and raising children. After, they came to my world, everything become much meaningful. They motivated me with their love and smiles. All were thanks to their existence in my life. My daughters are the toughest and most joyful chapters of my life.

My academic success wouldn't be possible if I didn't have the most amazing husband in my life. My dear husband, Bora Yıldız; you were the biggest supportive of this work. With every decision, you were with me, this means a lot. Thank you for being my best friend and love of my life for twenty years.

Thesis is dedicated to my husband, Bora Yıldız, who takes care of me and his family more than himself.

July, 2014

Tuğba YILDIZ

TABLE OF CONTENTS

LIST OF SYMBOLS

$bi_{t,d}$	Binary weight of term
C	A set of terms/concepts/vocabulary
$c_i \in C$	A term/concept
$<c_i,c_j> \in R$	An untyped semantic relation
$<c_i,t,c_j> \in R$	A typed semantic relation
$count(w_i)$	Number of $word_i$
d	Document as a vector
df_t	Document frequency of a term
$D(c_i)$	Depth of each concept
f_i	Feature vector
F	Feature matrix
$f_i \in F$	Feature vector in feature matrix
$gloss(r(c_i))$	A set of possible WordNet relations with gloss
IDF_t	Inverse document frequency
log-tf	Logarithmic term frequency
N	Dimension size
N_{00}	N_{00} is number of times neither x nor y occurs
N_{01}	N_{01} is number of times y appears without x
N_{10}	N_{10} is number of times x appears without y
N_{11}	N_{11} is the number of times x and y co-occur
N(S)	A set of the neighbors of S
P(c)	Probability of a concept
P	Patterns
$r \in R$	A relation in set of semantic relations
R	A set of semantic relation
S	A synset
S30	Success rate for 30 candidates
$t \in T$	A semantic relation type
t_i	i^{th} term
T	Semantic relation type
$TF_{t,d}$	Number of t in document d
$TF_{t,d} \times IDF_t$	Term frequency-inverse document frequency
$\|v\|$	Size of vocaublary
w_i	Term/word
x	Vector of x
x_i	An element of vector of x

y	Vector of y
y_i	An element of vector of y
#ofC	Number of cases
#ofCpW	Production capacity
#ofC>1	Number of cases whose whole are seen more than 1 times
#ofCpW>1	Number of cases per whole whose frequency is greater than 1
#W	Number of whole
#ofW>1	Number of whole whose frequency is greater than 1

LIST OF ABBREVIATIONS

a-scoring-f	Abstract Scoring Function
AVG_cpr	Average Case Per Row
AVG_cpc	Average Case Per Column
BalkaNet	Balkan WordNet
BOUN	Boğaziçi University
BP	Bootstrapped Pattern
CHL	Candidate Hyponym List
CI	Component-Integral
DAP	Doubly-Anchored Pattern
DFEAT	Distributional Features
DFEAT-PAT	Distributional and Pattern-based Features
DSIM	Distributional Similarity
DSIM-PAT	Distributional Similarity and Pattern-based Features
ESL	English as Second Language
FN	False Negative
FP	False Positive
GenCor	General Corpus
GP	General Pattern
Gr-bin	Graph Scoring with Binary
Gr-co	Graph Scoring with Co-occurrence
HSO	Hirst and St-Onge
IC	Information Content
IG	Information Gain
IR	Information Retrieval
IS-A	Is-a Relation
JCN	Jiang-Conrath
LCH	Leacock and Chadorow
LCS	Lowest Common Subsumer
LESK	Lesk Method
LF	Lexical Functions
LIN	Lin Method
LSP	Lexico-Syntactic Pattern
MC	Member-Collection
MRD	Machine Readable Dictionary
NC	Noun Compunds
NewCor	News Corpus
NLP	Natural Language Processing
NP	Noun Phrase

PAT	Pattern-based Features
PMI	Pointwise Mutual Information
POS-tag	Part of Speech Tag
RES	Resnik
Sim-2nd	Average Similarity Score
Sim-bin	Simple scoring with Binary
Sim-co	Simple scoring with Co-occurrence
Sim-cos	Simple scoring with Cosine
Sim-dice	Simple Scoring with Dice
Sim-hyper	Similarity with Target Hypernym
SVM	Support Vector Machine
TDK	Turkish Language Association
TDK-P	Turkish Language Association Pattern
TN	True Negative
TOEFL	Test of English as a Foreign Language
TP	True Positive
VSM	Vector Space Model
Wiki	Wiktionary
WordNet	Lexical Database
WUP	Wu and Palmer
WWW	World Wide Web
X^2	Chi-square
XCES	XML Corpus Encoding Standard
XML	Extensible Markup Language

LIST OF FIGURES

Page

LIST OF TABLES

Page

ABSTRACT

CORPUS-DRIVEN SEMANTIC RELATIONS EXTRACTION FOR TURKISH LANGUAGE

Tuğba YILDIZ

Department of Computer Engineering

PhD. Thesis

Adviser: Assoc.Prof. Dr. Banu DİRİ

Identification of semantic relations is the core problem in many Natural Language Processing tasks. One of the important tasks is to build up ontology or to construct thesaurus/lexicon. The most popular and widely used lexical database, WordNet is developed by manually. So it is used as source and also comparable work for most of the studies. Although these types of lexicons are reliable and effective, their production can be troublesome and time-consuming in some cases. So acquisition of semantic relation automatically from large amount of electronic documents (corpora, dictionaries, newspapers, newswires, etc.) becomes more important.

In this study, automatic and semi-automatic acquisition system for acquisition of hyponym/hypernym, meronym/holonym and synonym relations are handled from large corpus in Turkish Langage for nouns. For this purpose, some sort of methods is proposed to realize the model.

The method for hyponym/hypernym relation relies on lexico-syntactic pattern and semantic similarity. Once the model has extracted the items using patterns, it applies similarity based elimination of the incorrect ones in order to increase precision. Second model is based on similarity based expansion in order to increase recall. Several scoring functions are within bootstrapping algorithm are applied.

For meronym/holonym, lexico-syntactic patterns are utilized and adopted again to a Turkish huge corpus. Two different approaches are proposed to prepare patterns; one is based on pre-defined patterns that are taken from literature, second automatically produces patterns by means of bootstrapping method. Pre-defined patterns are categorized into two clusters; General and Dictionary-based patterns. Once these patterns help the system to extract matched cases, it proposes a list of part-whole pairs depending on their co-occur frequencies. For latter, bootstrapping model takes manually

prepared unambiguous seeds to induce syntactic patterns and estimates their reliabilities. Then, system extracts pair instances then ranks them by instance reliability scoring. Additional, statistical selection is used on global data obtaining from all results of entire patterns, where global data refers to a whole-by-part matrix on which several association metrics such as information gain, T-score etc. are measured and compared. Finally, how these patterns and statistical method improve the system accuracy especially within corpus-based approach and distributional feature of words is evaluated.

For synonym relation, the main assumption is that synonym pairs show similar semantic and syntactic characteristics by the definition. They share same meronym/holonym and hypernym/hyponym relations. Contrary to synonymy, hypernymy and meronymy relations can be easily acquired by applying lexico-syntactic patterns to a corpus. Such acquisition might be utilized and ease detection of synonymy. Likewise, some particular syntactic relations are utilized such as object/subject of a verb etc. Machine learning algorithms were applied on all these acquired features. The first aim is to find out which syntactic and semantic features are the most informative and contributes most to the model. Performance of each feature is individually evaluated with cross validation. The model that combines all features shows promising results and successfully detects synonymy relation. Another model is proposed to extract synonym relation with using integration of some sort of sources such as WordNet, bilingual on-line dictionary and monolingual on-line dictionary.

The main contributions of the study is considered as being first major attempt for Turkish hyponym/hypernym, meronym/holonym and synonym identification based on corpus-driven approach for Turkish Language. Second contribution is to use integrated approaches such as pattern-based method with statistical elimination and expansion, bootstrapping patterns, etc. for extracting relations. Third contribution is to use multiple resources such as WordNet, mono/bilingual on-line dictionaries, etc. and to integrate them

Ke**y words:** Lexico-Syntactic Pattern, Pattern-based approach, Bootstrapping approach, Distributional Similarity, Hyponym/Hypernym, Meronym/Holonym, Synonym, Semantic similarity.

ÖZET

DERLEM TABANLI ANLAMSAL SÖZLÜK OLUŞTURMA

Tuğba YILDIZ

Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

Tez Danışmanı: Doç. Dr. Banu DİRİ

Anlamsal ilişkilerin çıkarılması, Doğal Dil İşleme uygulamaları için büyük önem taşır. Bu uygulamalardan biride ontoloji/sözlük oluşturmaktır. Günümüzde sıkça kullanılan WordNet, insanlar tarafından elle oluşturulan bir sözlüktür. Birçok çalışmaya kaynak olan WordNet, ne kadar güvenilir ve etkili olsa da zahmetli ve zaman alıcıdır. Bu yüzden anlamsal ilişkilerin büyük elektronik dokümanlardan (derlem, sözlük, gazete, etc.) otomatik olarak çıkarılması önemli hale gelmiş, örüntü-tabanlı, dağılım benzerliği, makine öğrenmesi algoritmaları ya da hibrit yöntemler kullanılarak çözümler sunulmuştur.

Bu çalışmada, tam ve yarı otomatik yöntemler kullanılarak, isimler için alt/üst, parça/bütün ve eş anlamlılık ilişkileri Türkçe dilinde, derlem kullanılarak çıkarılmaya çalışılmış ve birkaç model sunulmuştur.

Alt/üst kavram ilişkisi için sunulan metot, sözlük-yapısal örüntülere ve anlamsal benzerliğe dayanır. Örüntüler, derleme uygulanarak aday alt kavramlar çıkarılmıştır. Sonrasında ise kesinliği arttırmak için benzerlik ölçütleri kullanılarak eleme yapılmıştır. Anma değerini arttırmak için farklı bir model olan istatistik tabanlı genişleme yöntemi kullanılmış, farklı skorlama ve ağırlıklandırma fonksiyonları modele dahil edilmiştir.

Parça/bütün ilişkisi için, örüntü yaklaşımı kullanılmış ve Türkçe derleme uygulanmıştır. İki farklı örüntü yapısı kullanılmıştır. İlki literatürde daha önceden tanımlı olan örüntülerin Türkçe'ye çevrilmesi ile gerçekleştirilmiştir. Diğer ise önyükleme metodu ile otomatik olarak belirlenmiştir. Tanımlı örüntüler, Genel ve Sözlük tabanlı olarak iki sınıfa ayrılmıştır. Bu örüntüler derleme uygulandıktan sonra, çıkan durumlar üzerinden birbirleri ile kaç defa çıktığı bilgisi kullanılmıştır. Diğer metot da ise önceden belirlenen kelime çiftleri kullanılarak, derlemdeki örüntüler bulunmuş ve örüntülerin güvenilirliği

hesaplanmıştır. Güvenli örüntüler yardımıyla yeni çiftler bulunmuş ve kelime çiftlerinin güvenilirliği hesaplanmıştır. Bazı ölçütler (bilgi kazancı, T-score gibi) kullanılarak karşılaştırma yapılmıştır. Son olarak bu örüntülerin ve metodun sistem doğruluğunu nasıl geliştirdiği incelenmiştir.

Eş anlamlılık ilişkisi için kullanılan yaklaşım, eş anlamlı olan çiftlerin benzer anlamsal ve sözdizimsel karakterlere sahip olmasıdır. Bu çiftler aynı alt/üst ve parça/bütün ilişkilerini paylaşırlar. Eş anlam ilişkisini, alt/üst ya da parça/bütün ilişkisindeki gibi örüntüler kullanarak derlem içinden yakalamak Türkçe için zordur. Bu yüzden bağımlılık ilişkileri (nesne/özne, etc.) kullanılmıştır. Çalışmanın ilk amacı modeli geliştirecek sözdizimsel ve anlamsal özellikleri çıkarmaktır. Bunun için herbir özellik çapraz doğrulama yöntemi ile değerlendirilmiştir. Model, özelliklerin birleşimi ile başarılı sonuçlar vermiştir. Bu yaklaşıma ek olarak, WordNet ve tek/iki dilli sözlükler kullanılarak verilen bir kelimenin eş anlamlısı derlemden çıkarılmıştır.

Çalışmadaki en büyük katkı, Türkçe derlem üzerinde alt/üst kavram, parça/bütün ve eş anlamlılık ilişkisinin yarı ve tam otomatik olarak çıkarılmasıdır. İkinci katkı, WordNet, sözlük gibi birçok kaynağın adapte edilmesi ile oluşturulan birleşik bir modelin kullanılmasıdır.

Anahtar Kelimeler: Örüntü-tabanlı yaklaşım, Önyükleme-tabanlı yaklaşım, Bölüşüm-tabanlı yaklaşım, Alt/Üst Kavram, Parça/Bütün, Eş Anlam, Anlamsal Benzerlik

CHAPTER 1

INTRODUCTION

1.1 Literature Review

Semantic relation refers to the relation between words, phrases, sentences and documents. In literature, comprehensive studies of semantic relation can be obtained with different perspectives [1], [2], [3]. For Lyons [1], who is a popular English linguist, "As far as the empirical investigation of the structure of language is concerned, the sense of a lexical item may be defined to be, not only dependent upon, but identical with, the set of relations which hold between the item in question and the other items in the same lexical system." Cruse [3] as a linguistics, give another description about semantic relation "the meaning of a word is fully reflected in its contextual relations; in fact, we can go further and say that, for present purposes, the meaning of a word is constituted by its contextual relations." According to Chaffin and Hermann's perspective [4] as psychologists, "semantic relations between concepts are basic component of language and thought".

Recently, semantic relation became major interest of computational linguistics. Various studies have been proposed for automatically identification of semantic relation from corpus. Most of the previous studies have been based on a key insight by Hearst [5] that lexico-syntactic patterns (LSPs) found in plain text to identify particular semantic relations. Other corpus-based attempts have used the statistics of co-occurrence and proposed bootstrapping mechanism [6]. In addition, distributional similarity techniques are utilized for constructed thesaurus [7]. Recently, machine learning algorithms are applied to syntactic, lexical and grammatical features that are obtained from corpus [8]. All these techniques can be employed together to develop integrated systems and increase the accuracy rate.

1.2 Objective of the Thesis

Identification of semantic relation from raw text is an important problem in Natural Language Processing (NLP). Numerous studies have been devised for extracting semantic relations. Most of them have been worked on English. Although valuable Turkish studies have been done in literature, number of studies is very few and based on dictionary definitions. This study is first major attempt based on corpus-driven integrated approach for Turkish domain.

The study aimed to develop an integrated model for acquisition of particular semantic relations; hyponym/hypernym, meronym/holonym and synonym from large Turkish corpus automatic and semi-automatically. The proposed model relies on combination of different approaches: lexico-syntactic patterns, distributional similarity and bootstrapping approach. All the techniques can retrieve semantic relations with promising results. The objective is to get better relevance and more precise results.

1.3 Hypothesis

A broad variety of methods are utilized especially for English to extract semantic relations. All predefined and the most widely used approaches for extracting semantic relations can be applied into Turkish domains. We realized this hypothesis with developing an integrated model to harvest particular semantic relations: hyponym/hypernym, meronym/holonym and synonym.

For this purpose, different approaches are adopted into proposed model. The most common approach is pattern-based approach that performed the hyponym/hypernym and meroym/holonym relations. Contrary to hypernym/hyponym and meronym/holonym relations, synonym relations can not be easily acquired by applying LSPs to a corpus. So another approach which is based on distributional similarity is carried out synonymy relations detection. In addition, corpus statistics are used with semantic similarity measurements to contribute the models.

CHAPTER 2

SEMANTIC RELATION

The widespread usage of World Wide Web (WWW) leads to enormous amounts of electronic text, including newspaper, emails, tweets, blogs, articles, documents from different domains, and so on. Browsing or filtering documents, extracting the information in which the people are interested, is an area of growing interest within Information Retrieval (IR) area. IR is one of the most important applications of Natural Language Processing (NLP), which is an interdisciplinary research area that deals with how computers can be used to understand and manipulate natural language text or speech. Beside information retrieval, other applications such as machine translation, question answering, and information extraction play an important role as NLP applications.

The diversity of approaches and applications are developed about knowledge levels of NLP; Phonetics and Phonology, Morphology, Syntax, Semantics, Pragmatics and Discourse. All these levels have been studied extensively in different perspectives such as computer science, linguistics, statistics, and mathematics and also concerned in other disciplines such as psychology, philosophy and anthropology.

Semantic is one of the knowledge levels in NLP and defined as “is the technical term used to refer to the study of meaning, and, since meaning is a part of language, semantic is a part of linguistics” [9]. Semantic relation is a subfield of semantic and refers to the relation between words, phrases, sentences and documents. Semantic information is valuable asset and essential for many NLP problems.

Semantic relation can be defined as a set of semantic relations R between a set of concepts C is a relation $R \subseteq C \times T \times C$, where T is a set of semantic relation types. A relation $r \in R$ is a triple $<c_i,t,c_j>$ that represent $c_i,c_j \in C$ with type $t \in T$.

It is possible to mention the existence of semantic relations' properties. Murphy [10] identified and clarified 8 properties: productivity, binarity, variability, prototypicality and canonicity, semi-semanticity, uncountability, predictability, universality.

1. Productivity: It is easy to produce new relations.
2. Binarity: Relations can binary that a word can relate one word.
3. Variability: Relations between words vary with the sense of the word and context.
4. Prototypicality and canonicity: Some word pairs present better relation examples and some word pairs seen as standard exemplars of a relation.
5. Semi-semanticity: Other properties which are non-semantic relations, such as grammatical category, co-occurrence, etc. have important role to determine whether a particular relation is considered to hold between two words.
6. Uncountability: The number of semantic relations is not counted and they are applied to open class.
7. Predictability: Semantic relations can be predicted from particular patterns and rules.
8. Universality: Semantic relations are general and same concepts are related with same semantic relations in any language.

As described above, uncountability defined as "the number of semantic relation types is not objectively determinable" [10]. When we observe the state-of-the-art lists of semantic relations used in the literature, there are contradictory views about number of semantic relations. Some studies [4], [11] proposed five semantic relations can be seen as primitive: class inclusion (hypernym/hyponym), part-whole (meronym), similars (synonym), contrast (antonym) and case relations. They provided a list of 31 semantic relationships [4] as sub-relations and these five families provide an apriori framework within other sub-relations. On the other hand, there is no agreement on the number and abstract level of semantic relations [12], [13], [14], [15]. In the context of this work, we focus on hyponym/hypernym, meroym/holonym and synonym relations. The most widely used semantic relations in literature are given below:

- **Hyponym/Hypernym**: Hyponym/Hypernym is a relation of inclusion and is known as IS-A relation in so many studies. e.g., "A dog is an animal", the term dog is a hyponym with respect to hypernym animal. Horizontal relation can be labeled co-hyponyms such as cat, bird and horse for animal. This relation is also

called subordination/superordination, subset/superset, generic-specific, taxonomy. Lyons defined hyponym/hypernymy relation as “the relation which holds between a more specific, or subordinate, lexeme and a more general, or superordinate, lexeme” [2].

- **Meronym/Holonym:** Another important semantic relation is the part-whole or meronym relations. Part-whole is a relationship between terms that respect to the significant parts of a whole. For example, “the eye is part of the face”, the term “eye” in the sentence is a part with respect to whole “face”. Cruse [3] describes the meronymy as follows “The whole-part lexical relation is an association between a lexical unit representing a part and a lexical unit representing its corresponding whole”. Palmer [9] uses term component for part-whole and defines component as “the total meaning of a word being seen in terms of a number of distinct element or components of meaning”.
- **Synonym:** Synonyms are words with identical or similar meanings. When two or more words have the same or nearly the same meaning in some or all senses, then they are synonymous. For example, “car is synonymous with auto”. Palmer identified synonym as “Synonym is used to mean sameness of meaning” [9]. For Chaffin, “terms that overlap in denotative meaning, connotative meaning, or both” [4].
- **Antonym:** Antonyms are words express opposite or incompatible meanings. For example, “fast-slow or old-young”. Chaffin described antonym as named contrast “this family consists of relations in which the meaning of one term contrasts, opposes, or contradicts the other term” [4].

Another discussion is about productivity property of semantic relations. Semantic relation is generally concerned with open-class words; nouns, verbs, adjectives, and adverbs. According to Miller [16], closed-class words generally play a grammatical role and open-class words play a referential role. The tangible reason for usage of them is that new or familiar open-class words can be used to express new concepts. For Miller, “Two facts about open-class words are immediately apparent: There are a great many of them and their meanings are intricately interrelated.” In addition to that, a new relation can be extracted whenever a new word is coined. It is clarified in Murphy’s [10] listing of semantic relations properties. Productivity, which is one of the properties of semantic relations, means that new relations among words can be created easily.

2.1 Semantic Relation between Nominals

Semantic relations have been a subject of other disciplines, more recently, is has become a major interest of computational linguistics. The studies in computational linguistics show a wide variety of methods of semantic relations on noun compounds especially in English. Noun compounds are “sequences of two or more nouns related through modifications” [17]. However the semantic classification of noun compounds (NCs) is seen as complex. Several reasons are listed as:

1. NCs have implicit semantic relations.
2. The interpretation of NCs is knowledge intensive.
3. It can represent more than one semnatic relations.
4. It is context-dependent.

Recently, studies are focused on extracting nominals (nouns) in the context of the sentences. One study produce the set of general rules, which are manually coded, are applied in order to interpret noun sequences in unrestricted text involving the taxonomic information [18]. Lauer proposed a set of 8 prepositional paraphrases: *of, for, with, in, on, at, about,* and *from* for compounds [19]. The study used corpus statistics with using computing frequencies of prepositions and involved them in probabilistic model. The other attempt classified noun compounds from the domain of medicine with using 13 semantic relations between the head noun and the modifier [20]. In the other study, Rosario et al. (2002) used the MeSH hierarchy and a multi-level hierarchy of semantic relations, with 15 classes at the top level to classify noun compounds [21]. Lapata classifies nominalizations “i.e. compounds whose head noun is a nominalized verb and whose prenominal modifier is derived from either the underlying subject or direct object of the verb" [22], [23] in domain independent text. The model is proposed as a statistical approach to interpret noun constituents. Nastase and Szpakowicz [13] dealt with noun phrases with a head noun and one modifier which can be noun, adverb and adjective. They extracted attributes of pairs from definitions with using WordNet and Roget's Thesaurus to capture relation between two elements. Two-level hierarchy with five semantic relations at the top and 30 semantic relation at the bottom was proposed. Then they used machine learning methods and similarity measurements to find the similarities. Other researchers [24], [25] used supervised and unsupervised learning algorithm for assigning semantic relation to noun-modifer pair.

Other comprehensive studies about identifying semantic relation especially on noun compounds proposed in recent years. Moldovan [14] proposed 35 semantic relations to classify in noun phrases. They identified feature vector of each noun phrases and then used semantic scattering to label noun phrases automatically [14]. Table 2.1 shows 35 different semantic relations that used in [14]. Instead of using primitive ones, they used mostly associative relation types. The same classes have been used with applying support vector machines (SVM) to classify semantic relations in nominalized noun phrases [26]. It was concluded that SVM seem better performance than other models. As a recent study, supervised, knowledge-intensive approach for the automatic semantic relation extraction between nominals was presented [8]. They used lexical, syntactic, and semantic features extracted from such as hand-built lexicon and additional annotated corpora.

Table 2.1 A list of semantic relations at various syntactic levels [14]

SR	**SR**	**SR**	**SR**
Possesion	Cause	Accompaniment	Probability of Existance
Kindship	Make/Produce	Experiencer	Possibility
Property/Attribute-Holder	Instrument	Recipient	Certainty
Agent	Location/Space	Frequency	Theme
Temporal	Purpose	Influence	Result
Depiction-Depicted	Source/From	Associated With	Stimulus
Part-Whole(Meronymy)	Topic	Measure	Extent
Hypernymy(IS-A)	Manner	Synonymy	Predicate
Entail	Means	Antonymy	

2.2 Related Works about Methods

Semantic relations are keys to various important particular NLP tasks such as information retrieval, information extraction, summarization, machine translation, question answering, textual entailment, and word sense disambiguation. Most of the studies rely on modern semantic resources such as thesauri, ontologies or lexical databases.

Taxonomy is a composition of a list of terms *C* organized into a hierarchy with a set of semantic relation *R*. On the other hand, thesaurus is a composition of a list of terms *C* organized into a hierarchical, equivalence and association relations *R* such as make/produce, cause, purpose, etc.Whereas lexical database is defined as “is a triple (*C, S, R*) where *C* is a vocabulary, *S* is a set of synsets, R is a set of semantic relations between S×T×S and T is set of semantic relations types” [28]. WordNet [29], [30],

[31], which is a large lexical database of English, is the most popular and useful resource to provide NLP applications. It includes entries for open-class words (nouns, verbs, adjectives and adverbs) that only organized into hierarchies. On the other hand, ontology is a general structure that is knowledge representation model [28].

Over the past decades, many considerable studies have dealt with semantic relation to create semantic lexicons. For this purpose, linguists made considerable efforts to collect information about words, find relations between them and build semantic resource such as lexicons, dictionary, ontology, Machine Readable Dictionaries (MRDs).

These types of resources such lexicons, thesaurus and dictionaries are reliable, effective and widely used. Whereas the process which is collecting and defining the terms, can be troublesome, time-consuming and cost of extension and maintenance operation is expensive in some cases. Because any lexicon should be updated in order to add new words. In addition to that, lexicon can be insufficient to cover domain specific words. For example, proper noun is highly important for some applications and WordNet's coverage of proper nouns is rather sparse.

In order to overcome these types of problems, NLP studies give more importance to ontology building, thesaurus construction, and semantic network construction automatically from some sources such as documents, corpus, Wikipedia, Web, etc.

According to Igo [32], techniques for building semantic lexicon can be divided into two groups: corpus-based methods and Web-based methods. "Corpus-based methods are typically designed to induce domain-specific semantic lexicons from a collection of domain-specific texts. In contrast, Web-based methods are typically designed to induce broad-coverage resources, similar to WordNet. Many domains use specialized vocabularies and jargon that are not adequately represented in broad-coverage resources (e.g., medicine, genomics, etc.)". With using these types of sources, many methods have been proposed, including pattern-based [5], statistical and bootstrapping methods [6], distributional similarity [7], knowledge-based methods [8], and machine learning techniques [15]. In addition, some studies which combined complementary approaches by looking for semantic relations.

The pattern-based method, which is leading one, is the most popular and widely used in the literature. The process of approach starts with defining which semantic relation will be involved and developing patterns that express that particular relation. Patterns are

searched in the sources to extract instances. These instances can be used directly or help to find new patterns recursively. For example, the LSPs such as “NPx is part of NPy” is used for meronym/holonym relation in studies. Instances that matched the pattern are used part-whole pairs or used to find new patterns.

Attempts based on patterns are proposed to extract lexical information from MRD in the literature [33], [34], [35]. Because of the limitations of MRDs, Hearst was the first to apply a pattern-based method to extract hyponym from unrestricted text, which is Grolier’s American Academic Encyclopedia [5]. Hearst’s approach became a pioneer and numerous studies used this approach for extracting semantic relations.

Pantel [36] presented *Espresso* algorithm which is a framework based on pattern-based approach in [5]. The method started with applying seed pairs to corpus and used generated sentences to exploit generic patterns. They scored the reliability of patterns and instances for filtering them. The top-10 best patterns were used to find new pairs.

Similar approach was also applied into on-line encyclopedia. One attempt addressed the problem of identification of semantic relations from Wikipedia [37]. They proposed an approach to identify lexical patterns in Wikipedia automatically and then they were applied to existing ontology, WordNet.

Another study [24] presented an algorithm based on Vector Space Model (VSM). Vectors were derived from statistical analysis that is obtained frequency of patterns of words by Web. These vectors were then used in a nearest-neighbor classifier.

One approach [38] for semantic relation extraction was based on combining LSPs and statistical techniques. LSPs were applied to corpus to detect a first set of pairs of co-occurrences. On the other track, statistical unsupervised system relies on distributional similarity, was used to obtain second set of pairs. Integration of both approaches was used for extraction semantic relations form corpus.

Various studies were also introduced other approaches for extracting, identifying or detecting semantic relations. A uniform approach [39] was described with using supervised corpus-based machine learning algorithm for classifying word pairs. Another approach was to build semantic lexicons for specific categories with simple bootstrapping mechanism with simple statistics [6], [40], syntactic information [41], [42], and improved version with LSPs [43]. A weakly supervised bootstrapping

algorithm that combined corpus-based method for inducing semantic lexicon with statistics of Web, was developed in study [31].

Several studies reported on corpus statistics approaches to noun compound. They used frequency of nouns and involved them in probabilistic model [19], [44], [45]. Some studies were based on hand-coded rules [17], [18]. There have been significant studies, which present supervised, semi-supervised, unsupervised, graph-based methods for automatic extraction of semantic relations. One study proposed pattern clusters method for nominal relation classification from large corpus in an unsupervised manner [46]. Another study presented an unsupervised method using graph model [41]. Another unsupervised method that held between nouns is based on discovering predicates that make explicit hidden relations [47]. Finally, a supervised, knowledge-intensive approach to the automatic identification of semantic relations between nominal was described [8]. Lexical, syntactic and semantic features were collected from different sources and a classification algorithm is applied.

Another way is using clustering algorithms on feature vectors to extract word senses [7]. This technique adopted the hypothesis that depends on the distributional similarity. Another important study [48] compared the knowledge-based, corpus-based and Web-based similarity measures for semantic relation extraction and reported which measures gives best results in which case.

2.3 Turkish Studies

For Turkish language, few studies have been presented for discovery of semantic relations. BalkaNet [49] was the first project to develop of a multilingual lexical database for Balkan languages such as Turkish WordNet. Although the project has not been completed yet, it was used for comparison with other studies.

One of the studies was proposed to construct Turkish WordNet automatically [50]. Four methods were proposed for automatic generation of Turkish WordNet: Translation from WordNet, Dictionary Definitions, Patterns, Usage of Unit Information. Two of them (Patterns and Translation) were applied to only hyponym/hypernym relations with 66% success ratio.

Another study was presented a rule-based method in order to extract semantic relations between words in a Turkish dictionary (TDK) [1] and to build a hierarchical structure as WordNet [51]. Rules in the study used surface form, category and definition of the word. They only applied the rules for hypernym and synonym relations. The success ratio is 94% for hypernym extraction. The hierarchy was compared with Turkish WordNet.

One of the recent studies to harvest semantic relations were based on TDK and Wiktionary (Wiki) [2]. They defined some phrasal patterns that are observed in dictionary definitions to represent particular semantic relations [52]. The accuracy rate of the prior relations: hyponym/hypernym (94%), meronym/holonym (55%), synonym (88%).

In some recent works, similar approaches were employed to develop a semantic network by using structural and string patterns in TDK [53], [54]. Relations used in studies were hyponym, synonym, antonym, member-of, amount-of, group-of and has-a. The overall accuracy is 86% for both studies.

Most of previous studies in Turkish depend on dictionary definitions and phrasal patterns. In the context of this study, an integrated model was developed for acquisition of particular semantic relations; hyponym/hypernym, meronym/holonym and synonym from large Turkish corpus automatic and semi-automatically. This study is first major attempt based on corpus-driven integrated with pattern-based and distributional similarity approach with using statistical measurements and other features that are obtained from mono/bilingual on-line dictionaries and WordNet. Antonym and case relations are eliminated due to the scope of the thesis is concerned with only nouns. Antonym relations are mostly between adjectives and case relations are generally between verbs.

[1] Türk Dil Kurumu (The Turkish Language Association).

[2] Vikisözlük: Özgür Sözlük.

CHAPTER 3

EXPERIMENTAL SETUP

3.1 Corpus

In our experiments, we used the BOUN Web corpus and language resources. They propose a set of language resources for Turkish language processing applications. They present an implementation of a morphological parser based on two-level morphology, an averaged perceptron-based morphological disambiguator with accuracy of 98%, and a Web corpus [55].

The BOUN Web corpus contains four sub-corpora. Three of them named NewsCor are from three major Turkish news portals and the other corpus named GenCor is a general sampling of Web pages in the Turkish Language. For encoding of the xml files, XML Corpus Encoding Standard (XCES) is used. The corpus is tokenized and encoded in paragraph and sentence levels and other symbols are also tagged. The size of the corpus is about 490M tokens.

3.2 Preprocessing

We conducted several experiments on unparsed corpora, and also we parsed it with the morpohological parser [55]. While the words are chosen, morphological disambiguator is included into system to select the less ambiguous word. As a result, each word in raw text is converted into surface form *surface+root+POS-tag.* For example, "*arabalar+araba+noun*" (cars+car+noun).

On the other hand, this form is insufficient in other experiments, especially for syntactic features extraction. Then we parsed each word into its morphemes. The representation

of a parsed tokens is in the form of *surface+root+POS-tag+[and all other markers]*.For example, "*arabalar+araba+noun+a3sg+pnon+gen*".

- +a3sg: 3sg number-person agreement
- +pnon: No possessive marker
- +gen: Genitive case marker

3.3 Methods

In this study, three different methods are used for extracting semantic relations. General procedures of each method are described with details in the next sections.

3.3.1 Pattern-based Approach

The most precise and well-known method that relies on LSPs is applied by Hearst [5] to raw text. The method is suggested for inferring the hyponym relations and it is stated that it is also available for other semantic relations. There have been so many attempts to extract semantic relation with using pattern-based approach. It is beneficial to find pairs and also for discovery of new patterns. General procedure of pattern-based approach is given for discovering new patterns/new pairs in the following:

1. Define the semantic relation (eg.hyponym/hypernym)
2. Collect a list of pairs (eg. apple/fruit) that are obtained by observed LSPs. The list can be automatically using the bootstrapping method.
3. Deploy the pairs to corpus
4. Find the patterns that indicate these pairs and keep them
5. Apply new patterns to second step again

3.3.2 Bootstrapping Approach with Seed Words

Bootstrapping method is commonly used in information extraction. Although the method is used in pattern discovery, it is also used without pattern-based approach. The approach is used for building semantic lexicon with initial seeds [6]. General outline of the algorithm is:

1. Choose categories (eg. vehicle) and small set of initial seed words for each category (initial seeds: car, auto, truck, plane, train)
2. Collect the context with window including seed words
3. Count the co-occurrence of words and compute the score of each word
4. Results are ranked and top-N words are selected as a new seed word
5. Return to step 2 and iterate n times

3.3.3 Distributional Similarity Approach

Distributional similarity approach, which is a popular method, is based on distributional hypothesis [56] which adopts that semantically similar words share similar contexts. The process of this approach was as follows; co-occurrence, syntactic information, dependency relations, etc. of the words surrounding the target word are extracted as a first step. This step can be named as feature extraction. Afterwards target word is represented as a vector with these contextual features. At the second step, the semantic similarity of two terms is evaluated by applying a similarity measure between their vectors. The words can be ranked according to their scores. Finally, top candidates are selected as most similar words from ranked list.

3.4 Similarity Measurements for Word and Vector Similarity

Similarity measure is a function that calculates a score from obtained feature vector. On the other hand, semantic similarity measures are used to evaluate similarity or relatedness of terms.

Methods that have been explained previously in Section 3.3 utilize some measurements to improve performance and compare the methods, respectively. In this study, various metrics are used to measure of word and vector similarity while extracting each semantic relation. Two basic algorithms are used to find word similarity: thesaurus-based and distributional algorithms (also vector similarity).

3.4.1 Thesaurus-based Methods

Thesaurus-based algorithms benefits from the structure of existing thesaurus and ontology to measure the semantic similarity/relatedness of terms. In this study, WordNet is used to compute only noun-noun similarity. For this purpose,

WordNet::Similarity package [57] is used. WordNet::Similarity package is freely available software package which covers modules for semantic similarity and relatedness between a pair of concepts. It includes six similarity measures and three relatedness measures. All are based on WordNet lexical database. WordNet::Similarity can be utilized by the utility program *similarity.pl*. It allows running all measures interactively.

For example, ***lin*** is one of the modules in WordNet::Similarity package and it relies on method represented by Lin [59]. A user can run *lin* module with two words pair such as car - bus and car - auto as following:

similarity.pl --type WordNet::Similarity::lin car bus

car#n#1 bus#n#1 0.603649218135011

similarity.pl --type WordNet::Similarity::lin car auto

car#n#1 auto#n#1 1

car#n#1 refers to the *first* WordNet *noun sense* of car associated with a word or word#pos combination. 0.603649218135011 represent the relatedness value between *car and bus* when using *lin* module.

In this thesis, ten modules are used to decide for similarity and relatedness. Three of them are based on the information content (IC) of the least common subsume (LCS). LCS of concepts c_1 and c_2 is the lowest node in the hierarchy that subsumes both c_1 and c_2. In the formula, words(c) is the set of words subsumed by concept c and N is the total number of words in the corpus. IC is a measure of the specificity of a concept and defined as,

$$P(c) = \frac{\sum_{w \in words\ (c)}\ count\ (w)}{N} \tag{3.1}$$

$$IC(c) = -\log P(c) \tag{3.2}$$

Three modules depend on IC and LCS in WordNet::Similarity package include of Perl modules that described in the following:

RES: The method is described by Resnik (1995) [58]. The method relies on the IC of LCS of two nodes.

$$Sim_{RES}(c_1,c_2) = -\log P(LCS(c_1,c_2)) \tag{3.3}$$

LIN: The method is described by Lin (1998) [59]. The method relies on the IC of LCS with sum of the IC of both c_1 and c_2.

$$Sim_{LIN}(c_1,c_2) = \frac{2\times logP\ (LCS\ (c_1,c_2))}{logP\ (c_1)+logP\ (c_2)} \quad (3.4)$$

JCN: The method is described by Jiang and Conrath (1997) [60]. The method relies on IC of LCS with sum of the IC of both c_1 and c_2. JCN takes the difference of the sum and IC of LCS.

$$Sim_{JCN}(c_1,c_2) = \frac{1}{2\times logP\ (LCS\ (c_1,c2))-(logP\ (c_1)+logP\ (c_2))} \quad (3.5)$$

The other three modules depend on path length in WordNet::Similarity package include of Perl modules that described in the following:

PATH: It is a baseline algorithm that represents the shortest path in the thesaurus between two concepts, c_1 and c_2.

$$Sim_{PATH}(c_1,c_2) = -\log shortestpathlen(c_1,c_2) \quad (3.6)$$

LCH: The method is described by Leacock and Chodorow (1998) [61]. It uses path with D that is the maximum depth of the taxonomy.

$$Sim_{LCH}(c_1,c_2) = \max[\ -\log shortestpathlen(c_1,c_2)/(2*D)] \quad (3.7)$$

WUP: The method described by Wu and Palmer (1994) [62]. It finds the depth of LCS and scales with sum of depths of the each concepts.

$$Sim_{WUP}(c_1,c_2)=2*depth(LCS(c_1,c_2))/(D(c_1)+D(c_2)) \quad (3.8)$$

There are four measures in the package as follows:

HSO: The method is described by Hirst and St-Onge (1998) [63]. HSO considers many other relations in WordNet and also consider other POS-tags.

Path_weight= C- path-length – (k * number of changes in direction) (3.9)

LESK: The method which is described by Banerjee and Pedersen (2002) [64], adopts the Lesk approach to WordNet. Relations are set of possible relations in WordNet whose glosses.

$$Sim_{LESK}(c_1,c_2)=\sum_{r,q\in Relations} \text{overlap}(\text{gloss}(r(c_1)),\text{gloss}(r(c_2))) \quad (3.10)$$

VECTOR: The method is based on word senses using second order co-occurrence vectors of glosses of the word senses. Context of pieces of text for Word Sense Discrimination is proposed [65]. This idea is adopted by [66], [67] to represent the word senses by second-order co-occurrence vectors of WordNet definitions.

VECTOR_PAIR: The module computes the relatedness of two word senses with using the VECTOR Algorithm. This measure is derived from [67].

3.4.2 Distributional Methods

Distributional method is clarified as "The intuition of distributional methods is that the meaning of a word is related to distributional of words around it" [68]. Because of limitations of Thesaurus-based methods such as lack of words, need of strong hyponym/hypernym relation etc., distributional methods can be used as complementary method. Distributional method represents features of context of word w. These context features of w can be extracted from corpus and obtains feature vector $f_i \in F$ matrix. Features can be collected from collocation, bag of words or dependency relations.

Collocation features captures the words that are positioned left or right of the target word w. Depending on the window size, root form of the word and POS-tags are used in the vector. Bag of words are unordered set of words of neighbors of target word w without importance of position. Dependency relations such as subject of, object of, etc. can be used as features under this assumption: "nouns bearing the same grammatical relation to the same verb might be similar" [68].

Co-occurrence is occurrence of two terms from a text corpus with in a broad context. It seems as good predictor for next word. Co-occurrence vectors can be derived by using co-occurrence statistics from large text corpora. The values between two words or a word and a feature can be measured with using some metrics. It is named as co-occurrence measures, weights or association measures.

3.4.2.1 Association Measures

Co-occurrence vector handles a value about its neighbor in its cell. It can be binary value like 1 if x and y occur in some context window, and 0 otherwise. Instead of binary value, usage of frequency or probability can be a better way. Since terms (y_1,y_2,…)

occur often word x are more likely to be good indicator. However they also are imperfect because of words that are appearing frequently such as the, and, they etc. So we need an association measures instead of raw count (frequency).

Information Gain (IG): is used to measure how many number of bits of information the presence or absence of a term in a document contribute to making the correct classification decision on a category. It is frequently used as a term goodness criterion in many problems in information retrieval. IG is also called expected mutual information, [68]. The formula of IG criterion of a term (c) and category (D) is defined to be in (3.11):

$$IG(c) = \sum_{i=1}^{m} P(D_i) \log P(D_i) + P(c) \sum_{i=1}^{m} P(D_i|c) \log P(D_i|c) + P(\tilde{c}) \sum_{i=1}^{m} P(D_i|\tilde{c}) \log P(D_i|\tilde{c}) \quad (3.11)$$

In the framework of the thesis, semantic relations are used in the formula instead of term and category. For example, while term represents “part” relation, category represents “whole”.

Pointwise Mutual Information (pmi): is the pointwise mutual information that is one of the commonly used metrics for the strength of association between two variables. Thus, it is worth analyzing the difference between information gain and mutual information. The formulas show that IG is the weighted average of the mutual information IG(c,D) and IG($\tilde{c}$,D), where the weights are the joint probabilities P(c,D) and P($\tilde{c}$,D). So IG is also called average mutual information. The main disadvantage of PMI is its bias towards low frequent terms. The pointwise mutual information criterion is defined in formula with x and y, which represent two words. N_{11} is the number of times x and y co-occur, N_{10} is number of times x appears without y, N_{01} is number of times y appears without x, N_{00} is number of times neither x nor y occurs, N is the total number of x. The formula is given in (3.12)

$$pmi(x,y) = \log \frac{N_{11}N}{(N_{11} + N_{01})(N_{11} + N_{10})} \quad (3.12)$$

Dice: It is very similar to pmi criterion. While pmi is theoretical measure, dice is empirical one. The dice coefficient of two sets is a measure of their intersection scaled by their size. The formula is defined to be as (3.13):

$$dice(x,y) = \log \frac{2N_{11}}{(2N_{11} + N_{01} + N_{10})} \quad (3.13)$$

Jaccard: It is the ratio of number of times the words occur together to the number of times at least any one of the words occur [69].

$$\text{Jaccard}(x,y) = \frac{N_{11}}{(N_{11} + N_{01} + N_{10})} \tag{3.14}$$

Chi-Sqaure (X^2): It measures the lack of independence between x and y using two way contingency table. The events A and B are defined to be independent if P(A,B)= P(A)P(B) or, equivalently, P(A|B) = P(A) and P(B|A) = P(B). The formula is defined to be:

$$X^2(x,y) = \frac{N(N_{11}N_{00} - N_{01}N_{10})^2}{(N_{11} + N_{01})(N_{11} + N_{10})(N_{10} + N_{00})(N_{01} + N_{00})} \tag{3.15}$$

T-score: It is defined as a ratio of difference between the observed and the expected mean to the variance of the sample. The formula is defined to be:

$$\text{T-score}(x,y) = \frac{(N_{11} - (N_{10} + N_{11})(N_{01} + N_{11}))}{N\sqrt{N_{11}}} \tag{3.16}$$

3.4.2.2 Vector Similarity Measures

Vector similarity measures are to find the similarity of two vectors, $\vec{x}$ and $\vec{y}$ which are represented as a vector. For the vector similarity, measurements of baseline, overlap, dice, jaccard, cosine, etc. can be used. dice and jaccard are also used in vector similarity. In this study, the most widely used measure that is cosine, is utilized.

Cosine: A word space in which, words are represented as vectors to compute similarity between two target words x and y. The co-occurrence can be measured with respect to documents, windows, sentences, or other units. The formula is defined to be:

$$\text{Cosine}(\vec{x},\vec{y}) = \frac{\sum_{i=1}^{N} x_i \times y_i}{\sqrt{\sum_{i=1}^{N} {x_i}^2}\sqrt{\sum_{i=1}^{N} {y_i}^2}} \tag{3.17}$$

3.4.3 Term Weighting Schema

The vector space model [70] is one of the most well known models that represent document and query as a vector in multidimensional term space. In vector space model, a document is represented as a vector in the term spaces, d= (w_1, w_2,…w$|V|$), where $|V|$,

is the size of vocabulary. The value of w_i between (0,1) displays how much the term contributes to the semantic of document.

Weighting is a way of numerical statistic which respects how important a word to a document. It is important to choose a proper weighting schema. There are various term weighting schema derived from the different assumptions and the probabilistic models including binary, raw count (frequency), logarithmic and inverse term frequency. Table 3.1 shows the weighting schema with formulas.

Table 3.1 Term weighting schemas

Term Frequncy Alternatives	**Formula**
Normal	$TF_{t,d}$
Binary	$bi_{t,d} = \begin{cases} 1, TF_{t,d} > 0 \\ 0, otherwise \end{cases}$
Logaritmic	$1+log(TF_{t,d})$

Term-Frequency ($TF_{t,d}$)**:** It is raw frequency and contrary to binary weighting, does matter how many times of a term appears in a given document d.

Binary ($bi_{t,d}$)**:** It is the simplest scheme and refers to absence or presence of a given term in related document; no matter how many times a term appears in a document. Thus, the possible values are either 0 or 1.

Logaritmic (*log($TF_{t,d}$)):* It is a smoothed frequency logarithmic $TF_{t,d}$ function. It is used to scale the effect of unfavorable high term frequency in a document.

When a document is considered, all the terms are equally important. Although some particular terms such as and, with, that, etc. (ve, ile, bu, vb.), generally appear so many times in a document, their effects are slight or "no discrimating power in determining relevance" [68]. Intervention is a need to weight of these terms. So that Document frequency df_t, defined to be the number of documents that contain a term t. Inverse document frequency (IDF_t) is defined to scale weight of t:

$$\mathrm{IDF_t} = \log \frac{\mathrm{N}}{\mathrm{df_t}} \tag{3.18}$$

TF-IDF: Combination of $TF_{t,d}$ and IDF_t is a statistical measure used to evaluate how important a word is to a document in a collection or corpus.

$$\mathrm{TF\text{-}IDF_{t,d}} = TF_{t,d} \times IDF_t \tag{3.19}$$

3.5 Performance Measures

The two most widely used measures for effectiveness of the system are precision and recall. Precision and recall have been used generally to measure the performance of information retrieval and information extraction systems. Recall indicates what proportion of all the relevant items have been retrieved from the collection. Precision indicates what proportion of the retrieved items is relevant.

Precision = #(relevant items retrieved) / #(retrieved items) = P(relevant|retrieved) (3.20)

Recall = #(relevant items retrieved) / #(relevant items) = P(retrieved|relevant) (3.21)

Table 3.2 Contingency table

	Relevant	**Not Relevant**
Retrieved	True Positive (TP)	False Positive (FP)
Not retrieved	False Negative (FN)	True Negative (TN)

According to Table 3.2, precision and recall can be represented as follows:

Precision = TP / (TP + FP) (3.22)

Recall = TP / (TP + FN) (3.23)

Another approach to combine recall and precision is the *F-measure*. The F-measure has been defined as a weighted combination of Precision and Recall. F-measure is shown as follows:

F-measure = (2*Precision*Recall) / Precision + Recall (3.24)

Alternative way is to use *accuracy* for evaluation the performance. According to Table 3.2, accuracy is represented as follows:

Accuracy = (TP + TN) / (TP + FP + TN + FN) (3.25)

In this study, precision, recall and F-measure are incorporated to evaluate the performance of the models. On the other hand, judgment of items as relevant or not in huge sized corpus is a basic problem. It is generally done manually by human as a gold standard which is used to judge the words indicate the proper semantic relation or not in this work. Three human annotators are manually tagged and evaluated the results.

CHAPTER 4

HYPONYM/HYPERNYM

The hyponym/hypernym relation is one of the semantic relations that play an important role in many NLP applications. The hyponym/hypernym relation referred as class inclusion, subclass/class, IS-A, a-kind-of, subordinate/superordinate, species/genus in the literature. Lyons defined hyponym as "the relation which holds between a more specific, or subordinate, lexeme and a more general, or superordinate, lexeme" [2]. In the words of Miller, "A concept represented by the synset {*x*, *x*', …} is said to be a hyponym of the concept represented by the synset {*y*, *y*', ...}, if native speakers of English accept sentences constructed from such frames as An x is a (kind of) y. The relation can be represented by including in {*x*, *x*', …} a pointer to its superordinate, and including in {*y*, *y*', …} pointers to its hyponyms" [16].

The hyponym/hypernym relation is tested by frames such as "*An X is a Y, An X is kind of Y"* or *"An X is type of Y"*. e.g., "A dog is an animal", the term dog is a hyponym with respect to hypernym animal. Cruse mentioned that the expression "*An X is a kind/type of Y*" is more discriminating than "*An X is a Y*" [71].

The hyponym/hypernym relation is generally seen as transitive and asymmetrical relation. There is a hierarchical structure between hyponym and hypernym. Horizontal relation can be labeled co-hyponyms such as cat, bird and horse for animal. All features of a hypernym are inherited to its hyponym. A study [72] defined that a hyponym inherits all features of hypernym with minimum one more feature that distinguish it from other co-hyponyms. For example, apple is a fruit and it inherits all features of fruit however it is different from orange under shape, taste, etc.

On the other hand, a study contradicts the transitivity of hyponym/hypernym with a following example [73]:

(a) A car seat is a type of seat

(b) A seat is a type of furniture

(c) A car seat is a type of furniture

The frames used for hyponym/hypernym relation is generally applied into nouns, not suitable for verbs. For example, "to amble is a kind of to walk" is not proper. There is no grammatical problem in the sentence but there is no general usage. Troponomy relation is a kind of entailment that refers as broader-narrower relations between verbs. Studies of troponomy are not part of this study because only nouns are considered in this study.

4.1 Related Works

Various studies are presented to acquire hyponym/hypernym relation automatically in recent years. Some studies employed LSPs, others used statistical, supervised, unsupervised, similarity-based approaches. All these techniques are applied into different sources such as dictionaries, corpus, Web, Wikipedia, etc.

Hearst was the first to apply LSPs for extracting hyponyms/hypernym pairs to text [5]. Following patterns are used in her study:

1. NP_0 such as {NP_1, NP_2, …, (and | or)} NP_n
2. Such NP as {NP, }*{(or | and)} NP
3. NP {, NP}*{,} or other NP
4. NP {, NP}*{,} and other NP
5. NP {,} including {NP, }*{or | and} NP
6. NP {,} especially {NP, }*{or| and} NP

First three patterns are built up by observation, other are derived from the aid of existing patterns by sketching a bootstrapping algorithm to learn more patterns from instances. All these patterns can be applied into raw text and pairs are extracted.

One of the studies [74] described a method that is supported by the Hearst's method to improve unsupervised ontology refinement algorithm by finding hypernymy patterns in domain-specific texts.

Mann [75] used part of speech patterns to extract a subset of IS-A relations involving proper nouns. Ruiz-Casado et al. [76] used WordNet to learn patterns for acquiring hyponymy relations with a precision of 69%.

Adopting pattern-based approach, Snow et. al. [77] built an automatic classifier for hyponym/hypernym relation. Firstly, predetermined hypernym pairs by WordNet were applied to training set for identifying large numbers of LSPs. They collected sentences that covers hypernym pairs and parsed them. They automatically extracted patterns from parse tree and combined these patterns. They created a classifier based on dependency path features to test whether a given noun pair is in the hyponym/hypernym relation or not. The best score showed 54% improvement over WordNet [77].

Ando et.al. [78] also used seven LSPs to Japanese Newspaper for hyponym extraction. They compared the results with associative concept dictionary. Another study again constructed a single query phrase such as "NPx is a/an NPy", and applied to search engine and evaluated extracted sentences [79].

Similar pattern-based approach [80] used newspaper corpus to construct a hypernym-hyponym based lexicon for Swedish automatically. Another study [81] used the same approach to create a hypernym-hyponym lexicon for Arabic with two modifications. First modification was to use only single pattern (NP_0 such as NP_1, NP_2...) that indicate IS-A relation. Second one was using the WWW to search for contexts of this pattern.

KNOWITALL was an unsupervised, domain-independent IE system that extracts information from the Web. Pattern-based approach was used and refined to increase recall [82]. Some of rules were adapted from Hearst's hyponym patterns and others were developed independently.

Several studies have been published on automatically identifying terms and their conceptual types from the Web or other huge corpus. One study [83] used Hearst's patterns [5] to learn semantic class instances and class groups. It applied hyponym patterns to the Web and acquired contexts around them.

Another pattern-based study for extracting hypernym pairs was applied to Web. They used a simple scoring function based on frequency to compute the hypernym evidence.

They used Dutch part of EuroWordNet for evaluation of the methods. Evaluation result showed the best precision value with 54% [84], [85].

Ritter et. al. [86] proposed three classifiers that are based on LSPs and corpus statistics to discover hypernyms: HYPERNYMFINDERfreq (based on frequency of Hearst pattern matches), HYPERNYMFINDERsvm (based on SVM classifier with additional features) and HYPERNYMFINDERhmm (based on Hidden Markov Model (HMM)) to extend the recall value. Recall was increased from 80% to 82% for common nouns with using HMM.

Shinzato and Torisawa [87] harvested hyponym relations from HTML documents. They used some important keys for HTML pages such as itemization, listing etc. Algorithm extracted hyponym candidates with using these types of keys and select them by using df_t and idf_t. And then they ranked and use semantic similarity. As a result they applied some heuristic rules to improve the accuracy. The score of precision was 61%.

Clustering approaches were also applied to IS-A extraction. Clustering algorithms grouped the words according to their meanings in text. Algorithm labeled them using its lexical or syntactic dependencies, and then extracted IS-A relation. Carabolla used conjunctions and appositives that appeared in the Wall Street Journal to build a hypernym-labeled noun-hierarchy like WordNet [88]. The idea was that nouns in conjunctions or appositives tend to be semantically related. Results were reported that 60% of nouns are evaluated with at least one hypernym. Recently, Pantel and Ravichandran [89] extended this approach with using syntactic dependency features for each noun with 81.5% success ratio.

Another approach [41] used graph structure to find semantic classes. The difference is that graph is based entirely on syntactic relations between words.

One of the important studies [90] represented a method by applying Latent Semantic Analysis (LSA) to filter extracted hyponyms and then used a graph-based model of noun coordination information to obtain promising results.

Another similar study [91] presented weakly supervised semantic class learning from the Web with using a single powerful hyponym pattern combined with graph structures. It provided a highly accurate semantic class learner that requires truly minimal supervision. Graph captured the ability of instances to find each other in a hyponym pattern rely on Web querying. For this purpose, they used the Doubly-Anchored Pattern

(DAP) to identify candidate instances for a semantic class. Extension of the same structure was used for term and relation extraction in another study [92]. Kozareva and Hovy [93] also proposed a semi-supervised method to learn and construct taxonomies based on Web. They used bootstrapping approach for hyponyms and hypernyms subordinated to the given a root concept and organized the concepts into a taxonomy structure.

A similar method for hyponymy acquisition is used relations from hierarchical layouts in Wikipedia. By using a machine learning technique and pattern matching, relations from hierarchical layouts in the Japanese Wikipedia were extracted. The precision was 76.4% [94]. Another attempt for acquiring hyponymy relations was an extension of the supervised method proposed by [95]. It differed in the way of enumerating hyponymy relation candidates from the hierarchical layouts and in the features of machine learning with improving the precision by 13.7%. Herbelot and Copestake parsed sentences in Wikipedia articles to obtain hypernym-hyponym pairs from the argument structures with a precision of 88.5% [96].

4.2 Methodology

In this study, two different models are proposed for acquisition of hyponym/hypernym relations for Turkish Language. Both integrated models rely on LSPs. Once the models have extracted the items using patterns and apply some elimination rules. And then each model built up the results with different ways. First model is based on elimination with using semantic similarity; the latter one performs similarity based expansion. Details of methods will be given in next section.

4.2.1 Candidate Hyponym/Hypernym Selection

The proposed model extracts hyponym/hypernym relations from a given corpus. For this purpose, the most precise and well-known acquisition methodology that is previously offered by Hearst [5] relies on LSPs, is applied. Starting with the same idea; four LSPs are selected and tested the results for Turkish as follows:

Pattern1: "NPs gibi CLASS" (CLASS such as NPs)

Example1: elma, armut, muz gibi meyveler (fruits such as apple, pear, banana)

Example2: elma gibi meyveler (fruits such as apple)

Pattern2: “NPs ve diğer CLASS” (NPs and other CLASS)

Example1: elma, armut ve diğer meyveler (apple, pear and other fruits)

Example2: elma ve diğer meyveler (apple and other fruits)

Pattern3: “CLASSlArdAN NPs” (NPs from CLASS)

Example1: meyvelerden elma, armut, muz (apple, pear, banana from fruits)

Example2: meyvelerden elma (apple from fruits)

Pattern4: “NPs ve benzeri CLASS” (NPs and similar CLASS)

Example1: elma, armut ve benzeri meyveler (apple, pear and similar fruits)

Example2: elma ve benzeri meyveler (apple and similar fruits)

The first pattern matched over 200,000 cases in our corpus from which 500 reliable hypernyms could be compiled, since that the hypernyms to be compiled have sufficient evidence in terms of number of occurrence. The second pattern matched only about 20,000 cases from which at most 100 hypernyms could be gathered. The last two patterns did not produce results judged satisfactory by manual observation. Once we compared the results of the patterns above, we further concluded that the first pattern is more accurate than the others and it successfully gives a strong indication of IS-A hierarchy. Given the syntactic pattern above, the algorithm extracts the candidate hyponyms that are recorded with their occurring frequency in the patterns. This count will be input to the scoring function for a later stage.

4.2.2 Elimination Based on Assumptions

The LSP defined above can extract many instances. However, some incorrect hyponyms are extracted due to parsing and other errors. Since the extracted list can contain non-hyponym words which are strongly associated with hypernyms, we need to filter them out. For example, looking at the word kuş/bird, the algorithm in first model (Model-1) extracted migration, photo, lake and other associated words along with the bird types. Also, unassociated words can be retrieved by mistake due to polysemy or parsing errors. The objective of this step is to exclude these kinds of non-hyponyms and to acquire more reliable candidates. According to our findings, the partial exclusion can be performed by making some simple assumptions. The assumptions which we applied for this step are as follows:

- In our most reliable pattern, “NPs gibi CLASS” (CLASS such as NPs), we observed that real hyponyms tended to appear in the nominative case and, therefore, without any suffix. We applied a rule that if a noun that appeared in the pattern “NPs gibi CLASS” was not in the nominative case but in the accusative, dative, or genitive case, it would be eliminated. For example, the frame “elmalar, armutlar gibi meyveler” (NPs are in plural form) is not accepted and used in Turkish. When we do not make this assumption, we observed system deterioration. This rule proved to be very effective, especially for an agglutinative language such as Turkish, since agglutinative languages have highly productive inflectional and derivational morphology.
- Secondly, we applied a rule based on the assumption that the more general a word is, the more frequent it is. A hypernym is assumed to be more likely to appear more frequently than its hyponyms. This rule is that, if a candidate hyponym has a higher document frequency (df_t) than its hypernym, it will be ignored. We tested the rule against manually crafted 1066 hypernym/hyponym pairs, out of which, 118 would have been eliminated by the rule. It means the rule works with an error rate of 10%. The assumption could only increase the precision, rather than the recall of the model.

4.2.3 Model-1: Statistical Elimination

Although we filtered out non-hyponyms based on the assumptions above, we can still have erroneous words. The objective of this step is to eliminate some of these words by relying on semantic similarity. Our expectation at this phase is that non-hyponym words share low semantic similarity with other candidates and hypernyms, while real hyponyms must have strong semantic relations with their class. The candidates with low similarity scores are likely to be erroneous. This semantic similarity of two words can be reduced to their frequency of co-occurrence in a corpus. The more frequently two words occur together, the higher their similarity is. For the similarity, the marginal total of the words must be taken into account.

In order to compute the similarity between concepts and eliminate incorrect candidates, we used the cosine similarity measurement based on word space model which is Vector Space as proposed in [65]. Schütze derives the word space model from the nearest neighbors of given word w_i in the corpus. In brief, all words can be described through

taking their co-occurrence relation with the words in a given window or a larger context. In a matrix, each row represents a word vector and each column represents a word. The $cell_{ij}$ records the number of times $word_i$ and $word_j$ co-occur together.

Dimension: For the Word Space model, the most significant task is how we determine the dimension and which words will be chosen. In our study, we applied the following selectional criteria. For a given hypernym, the words as the dimension of Word Space are derived depending on their semantic similarity with given hypernym or how they associate with it. The dimension words can be extracted from a global corpus or a local context. To build the dimension and to select the words of the dimension in this model, we mined BOUN corpus that is explained in Section 3.1.

The words can be ranked by their statistical measures of association. We calculated bigram values of word pairs from the corpora and compute their X^2 coefficient given in (3.15). In our experiment, we observed that the X^2 coefficient helps to gather the coherent words. Using the bigrams in which the target word occurs ranked by their coefficient, we can obtain plausible word lists which could be clue/key for acquisition of hyponyms. The nouns, the adjectives and the verbs are better potential indicators for understanding the meaning of the text than other POS-tags.

For the dimension, nouns, verbs and adjectives are chosen. Also some studies [87] used only verbs as their dimension. To balance the number of word types, we selected K number of words for each type. Finally, the most associated K number of neighbors is chosen as the dimension of the space. In this study, we selected K as 20 based on our observations.

A vector for a candidate hyponym is constructed from the nearest neighbor words which must be from our 60-word dimension obtained at the previous step. In brief we created a matrix in which the rows are vectors of each candidate and likewise each column represents a word from 60-word dimension. Each cell can refer to X^2 statistical score between a candidate hyponym and a word in the dimension. The target word (hypernym) is also added to the matrix along with these possible hyponyms to measure similarity between them.

Similarity Score: Briefly, all candidate hyponyms and the hypernym are represented in a candidate-by-word-list matrix. For this step, we need to measure similarities between

the hyponyms and the target word as well. The following procedure can be applied to measure the closeness of each candidate to the centroid or target hypernym word.

For each candidate c_i in Candidate Hyponym List (CHL):

$$\text{simple\_score}(c_i) = \sum_{j=1}^{N} c_{ij} \quad (4.1)$$

where CHL is a candidate hyponym list, c_i is an element of the list and N is the size of dimension. Since the dimension extracted the word depending on similarity to hypernym, the summation can be considered as semantic similarity to hypernym. And the candidates can be ranked by that score and eliminated.

On the other hand, we observed that cosine similarity ranking gives more precision and recall values than the summation procedure described above. Cosine similarity is a well-known measurement using vectors as follows: Once the similarities between candidate list (and target word as well), have been calculated, we get CHL X CHL up triangle matrix or symmetric matrix. Using these similarity scores, an average similarity score (or centroid) is calculated. Summation of a row or column gives the closeness of a concept to the centroid.

For the elimination phase, we denoted that average similarity score as *sim-2nd*, the similarity between the candidate and the target hypernym as *sim-hypernym* and the number of occurrence in LSP in Section 4.2.1 as *freq*. Finally we applied the following scoring function:

$$\text{score(cand)} = \begin{cases} \text{Pass, if } (\text{cand\_freq} > K1) \\ \text{Pass, if } (\text{freq} * \text{sim} - 2\text{nd} * \text{sim} - \text{hypernym} > K2) \\ \text{Fail, otherwise} \end{cases} \quad (4.2)$$

where K1 and K2 are specific thresholds for the domain. According to our observations K1=3 gives good performance. When the candidates appear more than three times in the patterns, they are more likely to be correct hyponyms and are automatically considered to have passed the test. In a list produced by LSPs for fruit, 20 out of 56 candidates retrieved from pattern matching at Section 4.2.1 occur more than 3 times and all of them are correct candidates. Moreover, most of the words can appear in the pattern by mistake due to error in data, polysemy or parsing error. Scoring function defined above works like a decision list. The instances are checked against conditions in the given order.

The first satisfied condition determines the output. So, if a candidate occurs less than 4, the defined formula will be checked, where *sim-2nd* and *sim-hypernym* weights are normalized. Thus, the score can be in the range [0-3]. According to the observation, K2 can be specified as 0.2. We have used only four classes: fruit, country, vegetable and fish. Finally, the candidates that have a poor score value and less frequent will be eliminated.

4.2.3.1 Experiments

We conducted several experiments on unparsed corpora, and parsed it with the parser as described in previous section. Each word in the raw text is converted into the form of surface/root/POS-tag.

We selected four hypernyms for the test phase; fruit, country, vegetable and fish. For a given hypernym, the algorithm searches the parsed corpora to match the pattern using regular expression. In order to calculate unigram, bigram information and statistical values, TextNSP [69] library is used. TextNSP that aids in analyzing n-grams in text using particular association measures such as the log likelihood ratio, Pearson's X^2 test, the dice coefficient, etc. For other calculations such as the pattern extraction process and second order representation, the necessary algorithms are implemented in the Java and Python programming languages. The algorithm is simply summarized in Figure 4.1 and Figure 4.2.

Algorithm 1. Dimension

```
NAME: getDIM
INPUT: C, H
OUTPUT: DIM
PURPOSE: For a given hypernym list, it outputs apropriate dimension of
Word Space Model

for each h in H
  list-bigram=retrieve bigram information for h in C
  ranked=rank list-bigram according to chi-square coefficient
  balanced-list=balanced-pos(ranked,20)
  return balanced-list
```

Figure 4.1 Create Dimension for Given Hypernym List (C=Corpus, H=Hypernym List, DIM=Dimension)

Algorithm 2. Producing Hyponyms

```
NAME: ProduceHyponym
INPUT: C, H, P
OUTPUT: Hyponym List for each h in H
PURPOSE: For a given hypernym list, it decides the hyponyms

for each h in H
    chl-inital= apply-pattern(h,P,C)
    chl-elim= applying-eliminatio-criteria(chl-initial)
    chl-by-word-matrix=  creatWordSpace(chl-elim,getDIM(h))
    chl-by-chl-matrix= cosine-similarity(chl-by-word-matrix)
    final-hyponym-list= eliminate(chl-by-chl, threshold)
    return final-hyponym-list
```

Figure 4.2 Create Hyponym List for each Hypernym in H (C=Corpus, H=Hypernym List, DIM= Dimension, P= Pattern, chl= candidate hyponym list)

The output of the first algorithm is used in the second algorithm. The second algorithm produces hyponym candidates for each hypernym in a given list.

4.2.3.2 Results and Evaluation of Model-1

In order to compare results, we used two different sources. Firstly, we used the BOUN Web Corpus and secondly, we searched the Web and manually found the list of a given target hypernym. Our examples were fruit, vegetable, fish and country. Once we have checked whether an item in the list appears in our corpus or not, we compared the results in a more realistic environment by checking against the Web list. Because of corpus-driven approach, we especially checked our resulting hyperyms against the list in the corpus, since the approach mines only the corpus. Table 4.1 shows what the possible size of the hyponym list in Web and corpus is as well. And it also shows us the number of retrieved items for each step.

Table 4.1 The number of items in Web, corpus and output of each step

Class	**Web**	**Corpus**	**S1**	**S2**	**S3**
Fruit	40	32	69	42	31
Vegetable	46	41	86	53	47
Country	189	137	1525	560	172
Fish	69	41	55	35	32

Looking at the first row, 40 items are retrieved from Web, 32 occurred in the corpus. There are 69, 42 and 31 items proposed by the Step1 in section 4.2.1, Step2 in section 4.2.2 and Step3 in 4.2.3, respectively. The precision/recall based evaluation method can

be used to analyze the results. Table 4.2 shows precision and recall values for each hypernym.

Two different recall values are evaluated. $Recall_1$ value shows the number of successfully retrieved items divided by the number of items existing in the Web. Similarly, to calculate the $Recall_2$, we divided the number of successfully retrieved items by the number of items in the corpus. Our algorithm uses only corpus data. The success is measured by looking at the $Recall_2$. On the other hand, $Recall_2$ represents capacity of the model and its performance ratio against the corpus.

Table 4.2 Precision and recall values for Fruit, Vegetable, Country and Fish

Hypernym	**Recall-Prec.**	**S1**	**S2**	**S3**
Fruit	$Recall_1$	69	66	58
	$Recall_2$	78	75	65
	Precision	44	71	84
Vegetable	$Recall_1$	88	76	76
	$Recall_2$	91	79	79
	Precision	52	70	79
Country	$Recall_1$	72	70	61
	$Recall_2$	96	95	83
	Precision	9	25	71
Fish	$Recall_1$	42	40	37
	$Recall_2$	61	58	54
	Precision	66	94	97
Average	$Recall_1$	68	63	58
	$Recall_2$	81	77	70
	Precision	69	65	83

As shown in Table 4.2, a significant increasing in precision values is produced by applying elimination at each step. For instance, the list in Step1 for Fruit hypernym includes some incorrect relevant items such as vitamin/vitamine or porsiyon/portion. In next step, portion remains but vitamine is eliminated. Finally, portion is eliminated in Step3 (Model-1). The second important result is that the decrease in recall is very small. Such a decrease is inevitable because we apply statistical elimination rather than statistical expansion in this model. After semantic similarity based improvement, the

average precision is increased by 20% and the recall values are decreased by 10%. The most significant enhancement is on country. Because it produces so many candidate hyponyms in Step1 however its actual value is less.

During the LSP phase, we observed that the more frequent items tend to be correct hyponyms. Therefore we applied a rule retrieving the candidates that occur at least 4 times. LSPs can be risky since incorrect hyponyms are often retrieved. But such incorrect candidates are mostly less frequent. So the challenge at this step is the elimination. The semantic similarity measurement is the main solution for the elimination problem.

Taking the result into account, the followings are our observations and findings;

1. More frequent items in the patterns tend to be correct hyponyms. Less frequent candidates in the patterns can easily be eliminated by making some simple assumptions. Similarity measurement is an efficient way to select correct hyponyms.
2. Corpus-based studies and algorithms suffer from data sparseness. For hyponym detection, there is big difference in the fraction of the siblings/hyponyms. For instance, while the hyponym “apple” appears in the patterns 30 times, only a few “kiwi” instances are found.
3. To create the dimension of word space, the algorithm can extract many unrelated words causing noise. Therefore, it is an important challenge to select the distinctive words and ignore misleading words. Moreover, the number of POS-tag (Noun/Verb/Adj) must be balanced as word space.

4.2.4 Model-2: Statistical Expansion

In the beginning of the Model-2, list of hyponym candidates are filtered by applying the LSPs and assumptions above and erroneous candidates may remain. To improve precision, we can take the candidates, sorted by their pattern frequency. The first K of these words can then be used as the original seeds for an expansion phase where K can be chosen experimentally (e.g. 5). Most studies [6], [40] selected K as 5.

The algorithm consumes the original seeds and expands them recursively adding new seeds one by one. The important factor here is to decide on a scoring function. Many approaches exist to select the most suitable candidate. We will explain our scoring

functions and algorithm in detail in the following sections. The algorithm will stop producing when it produces a number of items considered sufficient. The number to be considered "sufficient" may be automatically proposed by analyzing the capacity of the corpus.

Bootstrapping Algorithm: The algorithm is designed as shown in Figure 4.3. It first extracts hyponym/hypernym pairs and then applies bootstrapping with a scoring function. Where a-scoring-f denotes an abstract scoring function for selecting new hyponym candidates. Many scoring functions can be applied.

Generic Algorithm

```
Definitions:

INPUT: C, P
OUTPUT: list of
hyponym/hypernym pairs

Generate all hypernyms(H) and
their hyponyms based on P in C

for each h: H
 begin
  cand<-empty
  for each hyponym :hyponyms(h)
   begin
     if(pass the elimination
       criteria)
     cand  <- add hyponym;
   end
   seeds <-take first K cand;
   while (insufficient)
    add-new-one(seeds, a-scoring-f);
   store(h, final-seeds);
 end
```

Figure 4.3 The Generic Algorithm that Applies Bootstrapping Approach (C=Corpus, P= Pattern, H=Hypernym List)

Our scoring methodologies can be categorized in two groups: one group is based on a graph model; the other simply uses semantic similarity between candidates and seeds. We called the former *graph-based scoring* and the latter *simple scoring*. All scoring functions consume a list of seeds and propose a new seed.

Graph-Based Scoring: Graph-based algorithms define the relations between the words as a graph in a directed or undirected way. Each word is represented as a vertex and the relation between the words (or vertices) is represented as weighted edge. The study [41] proposed a similar approach and presented an incremental algorithm relying on a graph to build a hyponym cluster. Their method was very effective at avoiding infections stemming from spurious co-occurrences, polysemy and ambiguity.

Graph-based scoring is implemented as in Figure 4.4 in which each neighbor is compared not only with seed words but also with its other neighbors to avoid infections. To calculate edge score, there exist many weighting schema such as co-occurrence frequency, binary, dice, Jaccard, X^2, pmi, Cosine and so forth. We will define our detailed implementation in Section 4.2.4.1.

Graph-based Scoring

```
Definitions:
Let S a set of input seeds,
and N(S) be set of the neighbors of S.

INPUT: S
OUTPUT: new seed
for each n in N(S)
 begin
  for each m in N(S)
   begin
    if n != m
      score+= edge(n,m);
   end
   assign(score, n);
 end
 rank the N(S) by score
 return the best neighbor in N(S)
```

Figure 4.4 The Graph-based Algorithm (S= a set of input seeds, N(S)= a set of the neighbors of S)

Simple Scoring: This scoring method employs only the edge information between each candidate and the seeds. Therefore, the candidate which is the closest to the centroid of all seeds will be the winner.

As shown in Figure 4.5, the algorithm computes the similarity between a candidate and the seeds. Again the similarity functions between the concepts could adopt any weighting method; binary, idf, dice, X^2, pmi, or Cosine.

Simple Scoring

```
Definitions:
Let S a set of seeds, and
N(S) be set of the neighbors of S
INPUT: S
OUTPUT: new seed

  for each n in N(S)
   begin
    for each seed in seeds
     begin
       score+= edge(n,seed);
     end
    assign(score, n);
   end
  rank the N(S) by score
  return the best neighbor in N(S)
```

Figure 4.5 Calculate All Similarity Scores between the Candidates and Seeds. Then Return the Best Candidate. (S=a set of input seeds, N(S)=a set of the neighbors of S)

Edge Weighting: Both graph-based and simple scoring functions employ a similarity measurement to make a decision. Many weighting formulas have been used so far: dice, pmi, Jaccard, X^2, binary, idf, Cosine, kolmogorov, dissimilarity index and so forth. The measurements we analyzed in the study are as follows:

1. **IDF/co-occurrence:** The edge between the seed and the candidate can be calculated by multiplying co-occurrence by IDF for the candidate. In other words, co-occurrence is divided by the global term frequency of the candidate. Global term frequency is calculated by counting how many times a word occurs in a corpus.
2. **Binary:** If a seed and a candidate co-occur at least once in the corpus, the weight will be 1, if not, 0.
3. **Dice:** Measures how similar two seeds are in terms of the number of common bigrams.
4. **Cosine similarity:** To compute cosine similarity between the words, a word space in which words are represented as vectors is used. Each cell in a matrix contains the co-occurrence of $word_i$ and $word_j$.

For the cell value in a matrix, instead of using raw co-occurrence counts, other alternative weighting functions (such as log, dice) can be used to compute cosine similarity. In our study, although we used both raw and logarithmic weighting, we did not see any significant differences in the results in terms of system accuracy.

Building the Graph and Co-occurrence Matrix: The words whose similarities are to be measured can be represented in a matrix. $cell_{ij}$ represents the number of times $word_i$ and $word_j$ co-occur together. The matrix is a simple representation of a graph. Co-occurrence can be measured with respect to sentences, documents, paragraphs, or a given window of any size. The conventional way to compute co-occurrence is to use all neighbors within a window in a corpus by eliminating stop words. This approach has proved to be good at capturing sense and topical similarity [65]. For example, *train* and *ticket* can be found to be highly similar by this method. However, we need to apply more fine-grained methodologies to capture words sharing the same type such as *train* and *auto* or *ticket* and *voucher*.

To obtain such type of similarity, one solution would be representing words as vectors in modifier space rather than document or word space. In modifier space, $cell_{ij}$ represents the number of times that $head_j$ is modified by $modifier_i$. Nouns are similar to the extent that they are modified by the same modifiers [65].

Another solution is to use syntactic patterns to compute co-occurrence. To find nouns similar to a given list of noun seeds, we focused especially on nouns which share the same syntactic role in sentences. Nouns are considered similar when they are in particular patterns such as "N and N" or "N, N,..., N and N" (eg."elma ve armut" (apple and pear), "elma, armut, muz ve portakal" (apple, pear, banana and orange)). A similar approach was also used by [90]. Words considered similar would either all be subject, or all object or all indirect object. For example "John likes cake and coffee", cake and coffee would be represented by the system as a bigram, while the bigram "John and cake" would be rejected. This approach makes the model more fine-grained than other conventional ways of computing bigrams. It relies on the idea that those words are semantically closer to the extent that they co-occur within the particular patterns defined above.

4.2.4.1 Experiments

The model takes raw text then finds the structure (root, suffix, pos and other) of the words by using the Turkish morphological parser [55]. Each token in the corpus is represented in the form of surface/lemma/pos. The pattern “CLASS such as NPs” is applied to create initial seeds. Finally, the model statistically builds IS-A pairs.

We selected the most frequently occurring 17 out of 500 hypernyms that the LSP based module yields. Table 4.3 illustrates the selected classes and the first five seeds proposed for each by a pattern based methodology. For the methodology described in previous sections, we implemented a utility program in the Bash script and Java programming languages. The utility program can be used to verify and reproduce the results presented in the next section. We conducted several experiments. We tried LSP pattern for initial seeds and graph-based and simple scoring with various weighting functions for expansion. All are described as follows:

1. **Lexico-syntactic pattern (pattern):** After extracting instances, some candidates are eliminated using elimination conditions defined in the 4.2.1 and 4.2.2.
2. **Graph Scoring/binary (gr-bin**): All edges of the graph are weighted in a binary way. The edges will be 1 only if there is a co-occurrence relation between the words. If not, they will be 0.
3. **Graph Scoring/co-occurence (gr-co**): The edges of the graph are weighted by measuring of co-occurrence between words.
4. **Simple Scoring/binary (sim-bin**): Distances between words use binary weighting.
5. **Simple Scoring/dice (sim-dice):** Distances are weighted by the dice coefficient score between words.
6. **Simple Scoring/co-occurrence (sim-co):** Distances are weighted by the co-occurrence frequency between words.
7. **Simple Scoring/cosine (sim-cos):** The words are represented as vectors in a matrix. The cosine similarity between word vectors is used to weight edges.

Table 4.3 First 5 seeds for each category

Category	**First Five Seeds**
Tool Alet	knife, gun, cleaver, machine, telephone bıçak, silah, balta, makine, telefon
Bank Banka	Yaşarbank, Vakıfbank, Pamukbank, Esbank, Akbank Yaşarbank, Vakıfbank, Pamukbank, Esbank, Akbank
Device Cihaz	telephone, set, television, computer, printer telefon, set, televizyon, bilgisayar, yazıcı
Newspaper Gazete	Times, Milliyet, post, Cumhuriyet, Hürriyet Times, Milliyet, post, Cumhuriyet, Hürriyet
Illness Hastalık	cancer, aids, alzheimer, heart, flu kanser, aids, alzaymer, kalp, grip
Animal Hayvan	dog, cat, wolf, bird, horse köpek, kedi, kurt, kuş, at
Drink İçecek	tea, coffee, water, wine, alcohol çay, kahve, su, şarap, alkol
City Şehir	İstanbul, Ankara, İzmir, Bursa, Antalya İstanbul, Ankara, İzmir, Bursa, Antalya
Occupation Meslek	doctor, teacher, lawyer, engineer, expert doktor, öğretmen, avukat, mühendis, uzman
Fruit Meyve	orange, grape, watermelon, melon, apple portakal, üzüm, karpuz, kavun, elma
Mineral Mineral	calcium, potassium, phosphor, magnesium, ferric kalsiyum, potasyum, fosfor, magnezyum, demir
Event Organizasyon	championship, cup, concert, meeting, feast şampiyona, kupa, konser, toplantı, festival
Organization Örgüt	Kaide, Hizbullah, Hamas, Pkk, birlik Kaide, Hizbullah, Hamas, Pkk, union
Vegetable Sebze	carrot, tomato, cabbage, spinach, broccoli havuç, domates, kabak, ıspanak, brokoli
Sector Sektör	textile, tourism, food, automotive, agriculture tekstil, turizm, yemek, otomotiv, tarım
Sport Spor	basketball, volleyball, tennis, football, ski basketbol, voleybol, tenis, futbol, kayak
Country Ülke	France, Turkey, Germany, England, Russia Fransa, Türkiye, Almanya, İngiltere, Rusya

4.2.4.2 Results and Evaluation of Model-2

For the evaluation phase, we checked the proposed model against 17 selected hypernyms. In order to measure our success rate, we manually extracted all possible hyponyms of all the classes. To capture all hyponyms, once several functions have been run to produce as many candidates as possible and incorrect hyponyms were manually eliminated. We tested the methodology within the seven different settings described above. The pattern based procedure extracted a number of hypernym/hyponym pairs.

Then the other algorithms incrementally expanded the first five candidates produced by that pattern module and produced as many candidate hyponyms as pattern module produced. All settings produced the same number of hyponym candidates. We can see that pattern module outperforms other expansion algorithms at Table 4.4 in which, #_of_Output represent the size of output that produced by the pattern module, P is pattern module and Avg is average.

Table 4.4 Precision of the first experiment

Category	**#_of_ Output**	**P**	**gr-bin**	**gr-co**	**sim-bin**	**sim-dice**	**sim-co**	**sim-cos**	**Avg**
Bank	13	84	100	100	100	100	100	100	98
Mineral	12	91	100	100	91	100	100	100	97
Sport	27	100	92	92	100	92	92	96	95
Event	15	100	86	93	86	100	80	100	92
Profession	19	100	94	94	100	68	94	78	90
Illness	84	88	92	75	92	95	92	72	87
Animal	52	86	86	78	92	78	80	80	83
City	88	95	81	88	38	96	77	97	82
Fruit	32	90	78	81	71	75	93	50	77
Country	177	75	77	78	76	67	78	75	75
Device	22	95	90	59	86	59	68	50	72
Tool	23	86	65	69	56	73	65	60	68
Drink	18	88	66	66	61	72	50	38	63
Vegetable	33	93	54	57	48	45	78	57	62
Sector	69	88	52	57	62	44	57	56	59
Newspaper	21	90	52	42	57	47	61	61	59
Organization	26	76	30	53	30	61	61	19	47
Average	43	**90**	**76**	75	73	75	**78**	70	77

Pattern module and all the other expansion algorithms produced the same number of items. This number indicates the capacity of the pattern algorithm. The pattern algorithm is better than the others in terms of precision. In calculating recall, the size of the actual hyponym list must be taken into consideration.

In order to improve recall, we conducted a second experiment. In this experiment, the expansion algorithm consumes the first five candidates ranked and suggested by the pattern module, then expands the list to the size of actual hyponym list rather than the pattern capacity. When we maintain the size of the output as the size of the actual list for each hypernym, we get a better recall value as in Table 4.5 (#_of_output is equal to number of candidate hyponyms). We also observed that the ratio between the size of actual list and the pattern capacity is generally in the range (2.0-3.5). Depending on that range, the number of expansion iterations can be automatically determined. For

example, the number of iterations can be calculated by multiplying the pattern capacity by 2 or 3. Many studies constantly selected the iteration number as 50, 100 or 200. For a given hypernym, predicting how many hyponyms exist in the corpus would make more sense, but this would require another study.

Table 4.5 Recall analysis of second experiment

Category	**#_of_ Output**	**P**	**gr-bin**	**gr-co**	**sim-bin**	**sim-dice**	**sim-co**	**sim-cos**	**Avg**
Country	153	86	84	87	85	67	84	80	82
City	88	95	81	88	38	96	77	97	82
Mineral	42	26	80	88	80	80	80	80	73
Sport	50	54	76	74	78	62	74	78	71
Illness	145	51	73	70	75	71	77	57	68
Animal	85	52	76	54	77	62	71	49	63
Fruit	47	61	59	59	63	59	76	57	62
Bank	35	31	62	45	62	74	68	68	59
Event	55	27	54	58	54	50	61	47	50
Vegetable	54	57	55	42	44	44	50	44	48
Newspaper	32	59	46	28	50	34	50	53	46
Tool	50	40	46	40	46	46	57	42	45
Profession	92	20	54	55	54	39	54	38	45
Device	51	41	52	50	70	29	47	23	45
Drink	39	41	48	51	51	53	48	20	45
Sector	135	45	40	41	45	32	43	43	41
Organization	51	39	21	39	17	52	45	9	32
Average	71	49	**59**	57	**58**	56	**62**	52	56

As our third experiment, we incrementally altered the number of initial seeds which were produced by the pattern algorithm to investigate changes in recall. We used 10, 15, 20, 25, 30 and the pattern capacity as the initial seed size. The pattern capacity means that the expansion algorithms takes the whole of the output proposed by the pattern module as initial seeds and expand the list as many as the actual list for each hypernym. The average results are shown in Table 4.6.

The results indicate that increasing seed size gets better accuracy. This is because pattern module indeed gives promising results but is limited. Table 4.4 shows that the average score of the pattern is 90%. Since this accuracy is good, the expansion algorithms can simply and reliably exploit the outputs of the LSP algorithm as initial seeds. There is no significant difference between the accuracy of the different expansion algorithms. *gr-bin*, *sim-bin* and *sim-co* seem to be the best scoring functions. The graph-based algorithms and cosine similarity weighting are costly and time-consuming. We computed the bigram information and weighted our graph by using specific syntactic

pattern in a more fine-grained manner. It means only the words co-occurring in "N, N and N" pattern is accepted as bigram. Therefore *sim-co* or *sim-bin* which simply computes the relation is very successful. In Table 4.6, #_of_IS shows number of initial seed and All represent all output of the pattern is accepted as seed. As shown in Table 4.6, recall value of sim-co and sim-bin increased 71.6% and 72.5% when all output is used as seed.

Table 4.6 Recall analysis of third experiment

#_of_IS	**output_avg**	**pattern**	**gr-bin**	**gr-co**	**sim-bin**	**sim-dice**	**sim-co**	**sim-cos**	**Avg**
5	43	**89.7**	76.2	75.4	73.3	74.8	78.0	69.9	77.0
5	71	48.5	59.2	57.0	58.2	55.9	**62.5**	52.1	52.6
10	71	48.5	62.2	59.1	61.9	57.5	**64.2**	53.5	57.9
15	71	48.5	64.8	62.1	66.5	58.6	**66.9**	56.2	60.4
20	71	48.5	66.8	65.6	**67.4**	61.2	**67.6**	62.3	62.1
25	71	48.5	67.9	66.5	**68.6**	63.1	**69.3**	63.0	63.1
30	71	48.5	68.8	67.4	**69.9**	64.2	**70.1**	63.7	63.9
All	71	48.5	70.6	69.5	**72.5**	66.5	**71.6**	66.4	65.3

In this model, when looking at troublesome hypernyms having low accuracy in all tables, we faced a classical word sense problem. Organization, newspaper and sector are among the worst categories. For the newspaper hypernym we see a polysemy problem, e.g., nationhood (milliyet), independence (hürriyet), morning (sabah) and evening (akşam) are among the titles of the main newspapers in Turkey. All the titles have very distinctive meanings. Depending on the sense distribution, the expansion algorithm changes the direction of sense into frequently used senses. In this case, the usual sense of the terms is not the newspapers themselves but other non-hyponyms such as afternoon, night.

Our algorithm also suffers from collocations of compound words. If a preprocessing step is not applied, such factors deteriorate the model to produce incorrect candidates. If we had applied such a preprocessing phase, we would have had more success.

CHAPTER 5

MERONYM/HOLONYM

One of the important semantic relations is meronymy that represents the relationship between a part and its corresponding whole. The meronym is also mentioned in the literature with other reference such as part-whole, mereological parthood relations or partonomy [97], [98], [99]. Cruse defined that "the part-whole relation, in its lexical aspect, is called meronymy (sometimes partonymy)". Although same usage is seen in computational linguistics, it is indicated that usage of meronym and part-whole terms is not interchangeably and strictly correct. "Meronym is a relation between meanings whereas the part-whole is relation links two individual entities" [100].

The inverse relation of meronym is holonym that is defined "*Y is a holonym of X if X is part of Y*" [101]. For example, eye is meronym of face and face is a holonym of eye. Horizontal relation can be labeled co-meronymy such as mouth, nose and ear for face. At the same time, face is meronym of head and head is meronym of body. It terminates at body so that body is called as global holonym [97]. All the details about meronym relation will be given in the following sections.

5. 1 Related Works

Meronymic relationship has been a subject of some disciplines such as logic, philosophy, linguistics and cognitive psychology so far and it has become one of the major interests of computational linguistics. Logical and philosophical researchers are interested in the formal theory of part and whole that are based on "part-of" relation. It has been acknowledged as well in formal ontology [102], [103]. Part-of relation is defined as a strict partial-ordering, with the following axioms: existence, asymmetry, supplementarity, transitivity, extensionality, existence of mereological sum. In cognitive linguistics, a definition of meronym is attempted by [3] as follows "X is a meronym of

Y if and only if sentences of the form a Y has Xs/an X and An X is part of a Y are normal when noun phrases an X, a Y are interpreted generically". Cruse [97] also demonstrated the meronym relations with some test frames like "*A Y has Xs/an X*" eg. "A hand has fingers." However it is stated that this frame is too general like "A wife has a husband". Same problem occurs in second frame "*An X is part of a Y*". However the frames as "*The parts of a Y include/are X/Xs, Z/Zs etc.*" or "*The X and other parts of a Y"* do not leak. Cruse [97] also addressed the optionality (handle:door) or necessity (finger:hand) of the part-whole relation. Handle is an optional part for the door. On the other hand, finger is necessary part for the hand. Another discussion is about the distinction between parts and pieces and also mentioned in his studies. For example, "a glass jug dropped on a stone floor does not break up into parts, but into pieces." Differences can be summarized as parts have a distinctive function or they are separated from sister parts by a formal discontinuity.

Researchers on linguistics, psycholinguistics and cognitive psychology enlightened the nature of part-whole relation and meronymic relationships have long been recognized as being important. The investigation on part-whole relation is based on discussion of transitivity and different part-whole relations. Lyons [2] pointed out the question that has been debated in several studies, is whether meronymy relation is transitive or not.

In psycholinguistics, "part-of" relation is replaced by a family of relations. Types of meronym are emphasized in many studies [104], [105], [106]. Winston's definition of meronym relation is as follows: "One important type of semantic relation is the relation between the parts of things and the wholes which they comprise." Winston et al. [106] focused on the relation with expression such as "*The X is part of the Y"*, "*X is partly Y"*, *"X's are part of Y's"*, "*X is a part of Y"*, "*The parts of a Y include the Xs, the Zs..."*, and similar expressions. They answered the important questions about meronymic relations like "Are there several distinct families of meronymic relations or only one general type? How are meronymic relations to be distinguished from other semantic relations? And, are meronymic relations always transitive?"

In linguistics, Murphy [10] analyzed all fundamental semantic relations and meronym relation as well. It is defined as "Meronymy is the is-a-part-of (or has-a) relation, and (like hyponymy) the term refers either to the directional relation from whole to part or collectively to that relation and its converse, holonymy" [10].

In the words of Miller [28] who is the founder of WordNet which is a large lexical English database, “A concept represented by the synset {x,x’,. . . } is a meronym of a concept represented by the synset {y,y’,. . . } if native speakers of English accept sentences constructed from such frames as A y has an x (as a part) or An x is a part of y” [28]. Many definitions and aspects of the part-whole relation have been discussed in the literature however most of them relied on proposed studies that are given above.

5.2 Types of Meronym

Researches in linguistics, logic, and cognitive psychology have studied meronym relation and also often provided insights about the several different types of meronymic relations. Having many aspects of meronym relations turn out to be quite difficult and seems as a complex relation. There is no agreement on how to distinguish various kinds of meronymic relations. In many studies, the concept “part-of” relation was used to denote a family of meronymic relations. Because “part of” does not always refer to a specific meronymy. The problem with test frame such as “X is a part of Y” represents a variety of part-whole relations. Different types of part-whole relations have been proposed in the literature.

One of the most important and well-known taxonomies, designed by Winston [106] identified part-whole relations as falling into six major types of meronymic relations: component-integral object, member-collection, portion-mass, stuff-object, feature-activity, and place-area. They also analyzed types of meronym with relation elements: Functional, Homeomerous and Separable. The functional relational element means that parts have a functional role with respect to its whole (eg. handle-cup). Homeomerous parts are similar to each other and their wholes (eg. Slice-pie). Separable parts can be separated or disconnected from whole (eg. Steel-bike).

Component-Integral (CI) object relation is between components and the objects to which they belong. Objects may be concrete, physical, representational or abstract object, assemblies, organizations or the components of each of these types of things. Components are functional and separable. For example, pedal-bike, handle-cup are Component-Integral relation. Member-Collection (MC) is membership in a collection. Members refer to parts and they are separable from collection. For example, ship-fleet is an example of Member-Collection. Portion-Mass captures parts which are similar to each other and their wholes. Portion has no functional role however the parts are

separable and homeomerous because of similarity of part between each other and whole. For example, slice-pie is an example for Portion-Mass relation. Stuff-Object answers the question, "What is it made of?" Stuff-Object has no relational elements. "is partly" and "made of" frames are used to express the relation. For example, steel-car is Stuff-Object relation. Feature-Activity is a relation between parts as stages, phases, discrete periods, or sub-activities and whole as activities or events. Parts have functional role. For example, paying-shopping is an example of Feature-Activity relation. Place-Area relation is between areas and special places and locations within them. Every place within an area is similar to every other so it is homeomerous. For example, Oasis-Desert is Place-Area.

Another important attempt is mentioned by Iris et. al. [105]. They defined that the part-whole relation is as a collection of relations, not a single relation. They divided the meronym relation into four subtypes:

1. Functional component of its whole (functional component:whole/eg. engine:car)
2. The segmented whole (segment:whole / eg. slice:cake)
3. Member of a collection (member:collection / eg. sheep:flock)
4. Subset of set (subset:set / eg. fruit:food)

On the other hand, the most popular and useful ontology, WordNet, has also classified meronyms into three types: component-of (HAS-PART), member-of (HAS-MEMBER) and stuff-of (HAS-SUBSTANCE) [16]. Markowitz et al. [107] mentioned that part-whole relation appears in many semantic network models as a single link and it is divided at least four separate relations: funcomp (or functional component), member-set, subset-set (or is-a), and slice.

Some taxonomy [104], [108] based on the work of Winston et al. [106] have been proposed to define the semantics of the different part-whole relations types. Odell [108] introduced meronym as "Composition (also referred to as aggregation) is a mechanism for forming an object whole using other objects as its parts. It reduces complexity by treating many objects as one object." Gerstl and Pribbenow [104] also captured a classification of part-whole relations based on Winston [106]. Each class represents a different way of partitioning a whole into parts. Classification is listed as two parts: structure dependent parts and constructed parts. First level is the structure dependent relations that are isolated into three kinds of relations: Component-Complex, Element-

Collection and Quantity-Mass. Second level is the constructed parts that are divided into three: Segment, Piece and Portion.

Wanner [109] also studied based on Lexical Functions (LFs) and five different meronymic relations are captured by LFs:

1. Member-Collection (LF Mult). eg. Mult(vehicle)=fleet
2. Social Whole-Staff (LF Equip). eg. Equip(hotel)=reception
3. Organization-Its Head (LF Cap). eg. Cap(faculty)=dean
4. Whole-Its Uniform Unit (LF Sing). eg. Sing(sand)=grain
5. Whole-Its Centre (LF Centr). eg. Centr(mountain)=peak

Ontological aspects of the part-whole relation have been discussed [98], [110], [111]. They developed a formal taxonomy, which is based on well-known foundational ontology principles, to explain the semantics of part-whole relations. The formal taxonomy of types of mereological and meronymic part-whole relations is represented. The taxonomy is between transitive or mereological part-whole relations and their intransitive or meronymic counterparts.

5.3 Transitivity of Meronym

Semantic relations can have different properties and important one is transitivity. Given A, B, C are concepts and *R* is a semantic relation, the relation *R* is transitive if

$$[(A\mathit{R}B) \wedge (B\mathit{R}C)] \rightarrow (A\mathit{R}C) \tag{5.1}$$

Transitivity of meronymic relations is defined as “if A is part of B, and B is part of C, then A is part of C”. Among philosophers interested in axiomatic mereology, there is an almost complete consensus all parthood relations are transitive. However there are still no solutions for the practical use of transitive meronymy and arguments about existence of the transitivity of the part-whole relation come from linguists and cognitive psychologists.

Lyons [2] explained that part-whole relation is transitive if it refers physically discrete referent or referents are points or regions in physical space. For example, “handle:house” relation is acceptable because of physical entities. On the other hand, it

does not always appear to be true because the usage of transitivity in natural language is not compatible with logically transitivity. So that, it is concluded: “to say that part-whole lexical relations are non-transitive, rather than being all transitive or intransitive, is true enough; but it hardly advances our understanding of the structure of the vocabularies of language.” [2]

Two possible causes of intransitivity are discussed by [3]. Part-whole may be transitive or not depending on the context. Transitivity problem expressed in terms of functional domains: the handle is used to move the door by hand however the “handle” does not move the house so this it cannot be transferred to the house. Secondly the relation in (1c) can be seen as attachment relation instead of part-whole. To use “part of” as “attached to” is more appropriate in this example. Then the phrase “The house has a handle” is not acceptable because of the attachment of handle to the house. While the part-whole relation is transitive, the attachment relation is not.

(1a) The door has a handle

(1b) The house has a door

(1c) The house has a handle

This type of discussion raises questions about how many different part-whole relations exist. So transitivity and types of part-whole relation are parallel in many studies. Winston [106] claimed that meronym relation is transitive if same types of meronym relation are used in all phrases. For example, the term “part” is used as component-object sense in all phrases, and then it seems transitive. For the following sentences (2a-2c), all phrases in have component-integral object relation and relation between finger:hand:body is transitive.

(2a) Simpson’s finger is part of Simpson’s hand. (CI relation)

(2b) Simpson’s hand is part of Simpson’s body. (CI relation)

(2c) Simpson’s finger is part of Simpson’s body. (CI relation)

However, when different types of meronymic relations are used in the phrases, then the “*part of*” relation is not transitive. For example, (3a) is component-integral object type and (3b) is member-collection, then transitivity fails.

(3a) Simpson’s arm is part of Simpson. (CI relation)

(3b) Simpson is part of the Philosophy Department. (MC relation)

(3c) Simpson's arm is part of the Philosophy Department. (Neither CI relation nor MC relation)

Iris et. al. [105] supported that part-whole is not a single relation but a set of complex relations and divided into four subtypes: functional component:whole, segment:whole, member:collection, subset:set. These collections of four different part-whole relations with different transitivity behavior. Iris clarified the functional component:whole and member:collection are not necessarily transitive whereas segment:whole and subset:set are transitive.

5.4 Meronym and Other Related Relations

Confusion of meronymy relation with other semantic relations is a problem that is discussed in literature of semantic relations. Although meronym relation is easily confused with many other kinds of relationships such as attribution, attachment, and possession, it can be mostly confused with class inclusion (hyponym). Meronymy and hyponymy become intertwined in complex ways. The distinction and similarities between meronym and hyponym relation become a matter of some debates. Most of these debates express that it is not easy to distinguish them.

Hyponym/Hypernym is a relation of inclusion and is known as IS-A relation. The expressions like "*X is a Y*", "*X is a kind/type of Y*" are used generally to extract hyponym/hypernym relation. For example, "A dog is an animal", the term dog is a hyponym with respect to hypernym animal.

According to Winston, class inclusion and meronymy are distinguished when using "kind of" and "part of". For example, "a dog is part of animal" phrase is not correct [106]. Lyons also mentioned that class inclusion and meronymy are most difficult task to distinguish in the case of activities, although both relations are hierarchical relation that seems in thesauri, taxonomies and ontology [1], [2]. According to Cruse, meronymy relation is less straightforward than hyponym. To properly differentiate them is not an easy task. When two classes are given, there is no need for a separate remark of the relation of hyponym. However, it is necessary to make comment between two entities [97], [100].

Class inclusion is also easily confused with the member-collection relation because of involving membership of individuals [112]. Winston [106] also represented some psychological studies which have often included part-whole relations as examples of class inclusion relations. Hyponym seems as a subtype of part-whole relation in other study [105]. One study expressed that parts can be hyponyms as well as meronyms. “For example, (beak, bill, neb) is a hyponym of (mouth, muzzle), which in turn is a meronym of (face, countenance) and a hyponym of (orifice, opening)” [2].

Hyponym relation seems almost unproblematically transitive, however, there are some contradictory views about whether meronym relation is transitive or not. In many cases, transitivity seems to be limited. Although some studies [16], [99] represented both relations are asymmetric and transitive, Lyons [2] pointed out that meronym relation is non-transitive. Another discussion is about inheritance of meronym. Hyponyms are inherited from hypernyms whereas parts do not inherit features from whole [71].

In addition, the number of types for meronym relation is identified in different studies. Therefore, meronym relation seems more difficult because of distinguishing factors amongst meronymic relationships. “Part of” construction is not reliable for meronym relation or represents a general term which can be used to express various kinds of meronymic relations. For example, other relations such as attribution, attachment and ownership (or possession) can be confused with meronym relation. Although these relations can be seen as subtypes of meronym relation, they are treated as separate relations.

For attribution, the properties of an object can be confused with meronym. For example, the table object has a property such as height. While each table has a property of height, height is not part of a table. Attachment is often intertwined with meronym relation. For example, fingers are attached to hands. However, while earrings are attached to ears, they are not part of ears. Ownership is another relation that is confused with meronym. “has” frame is used to express meronym relation frequently. Distinction between ownership and meronym can be analyzed in real world. While “has” frame in (4a) express ownership, (4b) means wheels are parts of bicycle.

(4a) Ali has a book

(4b) Bicycle has wheels

5.5 Meronym Studies in Computational Linguistics

The part-whole relation between nouns is generally considered as a fundamental semantic relation. The discovery of meronym relations plays an important role in many NLP applications, such as question answering, information extraction [113], [114], [115], query expansion [116] and formal ontology [110], [117].

In computational linguistics, comprehensive list of studies has been done on meronym. Most of them are about automatically detecting part-whole relation. A variety of methods have been proposed to identify part-whole relations from a text source. Some studies employed LSPs which is a useful technique especially in semantic relation extraction. It is the most preferred method due to its simplicity and success. There have also been other approaches such as statistical, supervised, semi-supervised or WordNet corporation [36], [118], [119], [114], [120], [121].

Various studies for automatically discovering part-whole relations from text have been based on Hearst's pattern-based approach. Hearst developed a method to identify hyponym (IS-A) relation from raw text by using LSPs. Although the same technique was applied to extract meronym relations in [5], efforts were reported to be concluded with no great success.

In [118], a statistical method was proposed to find parts in very large corpus. Using Hearst's methods, five lexical patterns and six seeds (book, building, car, hospital, plant, school) for wholes were identified. Part-whole relations extracted by using patterns were ranked according to some statistical criteria with an accuracy of 55% for the top 50 words and an accuracy of 70% for the top 20 words.

A semi-automatic method was presented in [113] for learning semantic constraints to detect part-whole relations. The method picked up pairs from WordNet and searched them on text collection: SemCor and LA Times from TREC-9. Sentences containing pairs were extracted and manually inspected to obtain list of LSPs. Training corpus was generated by manually annotating positive and negative examples. The decision tree [122] was used as learning procedure. The model's accuracy was 83%. The extended version of this study was proposed in [119].

Van [115] developed a method to discover part-whole relations from vocabularies and text. The method followed two main phases: learning part-whole patterns and learning wholes by applying the patterns. An average precision of 74% was achieved.

A weakly-supervised algorithm, Espresso [36] used patterns to find several semantic relations besides meronymic relations. The method automatically detected generic patterns to choose correct and incorrect ones and to filter with the reliability scoring of patterns and instances. System performance for part-of relations on TREC was 80% precision.

Another attempt on automatic extraction of part-whole relation was for a Chinese Corpus [121]. The sentence containing part-whole relations was manually picked and then annotated to get LSPs. Patterns were employed on training corpus to find pairs of concepts. A set of heuristic rules were proposed to confirm part-whole relations. The model performance was evaluated with a precision of 86%.

Other important studies were proposed in [114], [120]. A set of seeds for each type of part-whole relations was defined. The minimally-supervised information extraction algorithm, Espresso successfully retrieved part-whole relations from corpus. For English corpora, the precision was 80% for general seeds and 82% for structural part-of seeds. In [120], an approach extracted meronym relation from domain-specific text for product development and customer services.

In Turkish, recent studies to harvest meronym relations and types of meronym relations are based on phrases that occur in dictionary definition such as (TDK) and Wiktionary [51], [53], [54]. All defined meronym relations in these studies are explained in next sections.

5.6 Methodology

Grammatical aspect of meronym relation becomes another discussion in literature [2], [97], [106], [123]. Some argued that nouns are more suitable than verbs for meronym. “If x and y are nouns, An x is a part of a y frame is acceptable” [97], [123]. Cruse’s definition of meronym covers the relation among nouns. According to Miller [28], verbs cannot be taken apart in the same way as nouns. On the other hand, other studies promoted availability of other POS-tags. Winston et. al. [106] claimed that any verbs that are nominalized, can be related to other nouns (eg. paying:shopping). They also argued that meronym differs from attribution so that usage of adjective reduces in meronymic relations. Murphy [10] proposed that non-nominal descriptions are not meronymic relation whereas nominalized description of activities or properties are

acceptable [10]. In the scope of this study, meronym relation is also considered as a noun-to-noun relation rather than other POS-tags.

In this study, we presented a model for semi-automatically extracting part-whole relations from a Turkish raw text. For this purpose, we evaluated three different clusters of patterns in different aspects; General Patterns (GP), Dictionary-based Patterns (TDK-P), and Bootstrapped Patterns (BP). First cluster is based on GP which are the most widely used in literature. These patterns are collected from some pioneer studies [98], [106], [119] and analyzed in Turkish. 240K cases are obtained from GPs. While GPs are widely used and well known especially within a huge corpus, the TDK-P is suitable and applicable to dictionary-like resources (TDK, WordNet, Wikipedia, etc.). Although the latter is suitable for dictionary, we discussed that it can have a capacity to disclose semantic relation even from a corpus. In this study, TDK-P is based on patterns that are caught from TDK and Wiki. The number of cases is 509K for TDK-P.

We adopted both types of patterns to extract the sentences that include part-whole relations from a Turkish corpus. Some patterns which are not suitable and applicable for Turkish language are eliminated. The most frequent wholes are selected for each LSPs. Each whole and its potential parts are ranked according to their frequencies. Third cluster is based on bootstrapping of the unambiguous set of part-whole seeds. Some manually prepared seeds are used to induce LSPs and score them. Six reliable patterns are extracted; some are eliminated according to experiments. We compared the strength of some association measures with respect to their precisions. Variety of statistical methods is applied on the global data obtained from the clusters to improve system performance. For the evaluation, we selected first 10, 20 and 30 candidates ranked by the association measures such as dice, T-score, IG, χ 2, etc. to evaluate their performance in the study. Statistical measurements are applied to a large data set obtained from all pattern results. They are compared in terms of precision and recall scores within a variety of experiments. The proposed parts are manually evaluated by looking at their semantic role.

5.6.1 General Patterns (GP)

The most precise acquisition methodology applied earlier by Hearst [5] relies on LSPs. We started with the same idea of using the widely used patterns, GP, to acquire part-whole relations, which are the widely used and well-known patterns from several

studies [98], [106], [119]. One of these studies was proposed by Winston et al. used frames as “part of”, “partly” and “made of” for six different types of meronymic relations. Girju et al. [119] represented that some patterns always refer to part-whole relation in English text, while most of them are ambiguous. Keet and Artale [98] developed a formal taxonomy, distinguishing transitive mereological (1) part-whole relations from intransitive meronymic (2) ones. All general patterns are listed in Table 5.1. Although there are also various studies that have used pattern-based approaches, most of them are subsumed by the following patterns.

Table 5.1 Patterns that are used in three different studies

Winston (1987)	**Girju (2006)**	**Keet (2008)**
NPx part of NPy NPx partly NPy NPy made of NPx	parts of NPy include NPx NPy consist of NPx NPy made of NPx NPx member of NPy One of NPy constituents NPx	NPx member of NPy (1) NPx constituted of NPy (1) NPx subquantity of NPy (1) NPx participates in NPy (1) NPx involved in NPy (2) NPx located in NPy (2) NPx contained in NPy (2) NPx structural part of NPy(2)

The patterns are manually adopted into Turkish equivalences as shown in Table 5.1, where syntactic and morphological difficulties are handled by suitable LSP with regular expressions. The patterns are equivalent to the English patterns in terms of translation and meaning. The process is carried out by accessing and utilizing each morpheme to extract the sentences bearing part-whole relation. As expected, some patterns which are not suitable and applicable for Turkish language are eliminated. The remaining patterns are evaluated in terms of capacity and reliability. Summary of general patterns are given in Table 5.2. Examples of each pattern in English and Turkish are shown in Table 5.3. Variation of patterns and Turkish equivalents are listed with details in Table A.1.

To extract the sentences which include part-whole relations by using LSPs from a Turkish corpus of 490M tokens, Turkish equivalents of these patterns are constructed in regular expression forms.

In order to evaluate the approach, we picked up the most frequent wholes for each LSPs. For each whole, its potential parts are ranked according to their frequencies. To distinguish the distinctiveness, we normalized frequency by dividing the number of times a part occurs with given whole by number of times a part retrieved by all patterns.

We selected first 30 candidates ranked by their scores for evaluation. The proposed parts are manually evaluated by looking at their semantic role.

Table 5.2 A summary for General Patterns (GP)

General Patterns	**#of Cases**	**#of Wholes**	**The most frequent wholes**
NPx part of NPy	19K	2.5K	Life, Culture, Turkey, Europe
NPx member of NPy	23K	2K	Commission,Turkey,Group,Union,Family
NPy constituted of NPx	598	293	System, Program, Project
NPy made of NPx	6.3K	1.7K	Questionnaire, Public opinion, Publication
NPy consist of NPx	9.2K	2K	Report, Material, Product, Food
NPy has/have NPx	120K	8.2	Turkey, Person, Job, Government, Team
NPy with NPx	68.8K	8.7K	Person,Government,Turkey,Kid,Woman, Patient

Table 5.3 Examples of GP

General Patterns in English and Turkish	**Examples in English and Turkish**
NPx part of NPy NPx NPy (bir|-) parçasıdır	Nose is part of face Burun, yüzün bir parçasıdır
NPx member of NPy NPx NPy (bir|-) üyesidir	Germany is a member of AB Almanya, AB'nin bir üyesidir
NPy constituted of NPx NPxNPy (bir|-) bileşenidir	Program is constituted of input/output Giriş/çıkış, programın bir bileşenidir
NPy made of NPx NPy NPx'DAn yapılır	Cake is made of egg Kek, yumurtadan yapılır
NPy consist of NPx NPy NPx içerir	Fruits consist of vitamine Meyveler vitamin içerir
NPy has/have NPx NPy NPx vardır	Team has a captain Takımın kaptanı vardır
NPy with NPx NPx olan NPy	Woman with glass Gözlüğü olan kadın

5.6.2 Dictionary-based Patterns (TDK-P)

The most efficient and reliable way of applying LSP is to extract information from Machine Readable Dictionaries (MRDs). The use of language in dictionary is generally simple, informative, and structured and highly includes a set of syntactic patterns. Thus, many studies have exploited the dictionary definition recently. For Turkish, the recent studies to harvest meronym relations used dictionary definition (TDK) and Wiki [51], [53], [54].

Table 5.4 A summary for Dictionary-based Patterns (TDK-P)

Dictionary-Based Patterns	**#ofCase**	**#ofWhole**	**The most frequent wholes**
Group-of (whole|group|all|set|flock|union of)	22.7K	3.6K	Game,Human,Woman, Football,Person
Member-of (class|member|team of)	20K	3.8 K	Turkey,Team, Newspaper,University
Member-of (from the family of Y)	184	47	Legumes,Rosaceae, Citrus fruit
Amount-of (amount|measure|unit of)	3.4K	1.4K	Bank,Dollar,Euro, Turkey
Has/Have (Y has the suffix of l(H))	445K	13.7K	Human,Woman, Football,Person
Consist-of	12.4K	2.7K	Group,Committee,Team Exhibition,Book
Made-of	4.9K	1.4K	Payment,Interruption, Import

Table 5.5 Examples of TDK-P

Dictionary-based Patterns in English and Turkish	**Examples in English and Turkish**
Group-of (whole|group|all|set|flock|union of) NPy whole of NPx NPy NPx bütünüdür	Democracy is whole of laws Demokrasi, kanunlar bütünüdür
Member-of (class|member|team of) NPy team of NPx NPx NPy takımıdır	Drill and hammer is team of tool Matkap ve çekiç alet takımıdır
Member-of (from the family of Y) NPx from the family of NPy NPy gillerden NPx	Bean is from the family of leguminous Baklagillerden fasulye
Amount-of (amount|measure|unit of) NPx unit of NPy NPx NPy birimidir	Disk is a unit of storage Disk, depolama birimidir
Has/Have (Y has the suffix of l(H)) NPy with NPx NPx(lI) NPy	Woman with glass Gözlüklü kadın
Consist-of NPy consist of NPx NPy NPx içerir	Book consist of chapters Kitap, bölümler içerir
Made-of NPy made of NPx NPx yapılan NPy	Gloves are made of wool Yünden yapılan eldiven

In [53], semantic relations were extracted to build semantic network. In [51], they presented different automatic methods to extract semantic relationships between concepts using two Turkish dictionaries. They efficiently used regular expressions to extract part-whole relation. We examined all these findings and provided a summary

report for dictionary-based patterns as shown in Table 5.4. Examples of each pattern in English and Turkish are shown in Table 5.5. Details are in Table A.2.

Member-of, made-of, consist-of and has/have can be confused with the ones in the GP whereas pattern specifications are different from each other. All patterns are applied to Turkish corpus as in general patterns and a similar process is carried out. Even though these patterns are useful especially in dictionary, they are needed to check if they could return redundant and incorrect results or not for Turkish.

5.6.3 Bootstrapped Patterns (BP)

Methodology of bootstrapped patterns is totally different from that of others described above. The bootstrapped pattern approach proposed is implemented in two phases: Pattern identification and part-whole pair detection. Figure 5.1 represents how the system is split up into its components and shows data flow among these components. The system takes a huge corpus and a set of unambiguous part-whole pairs. It then proposes a list of parts for a given whole.

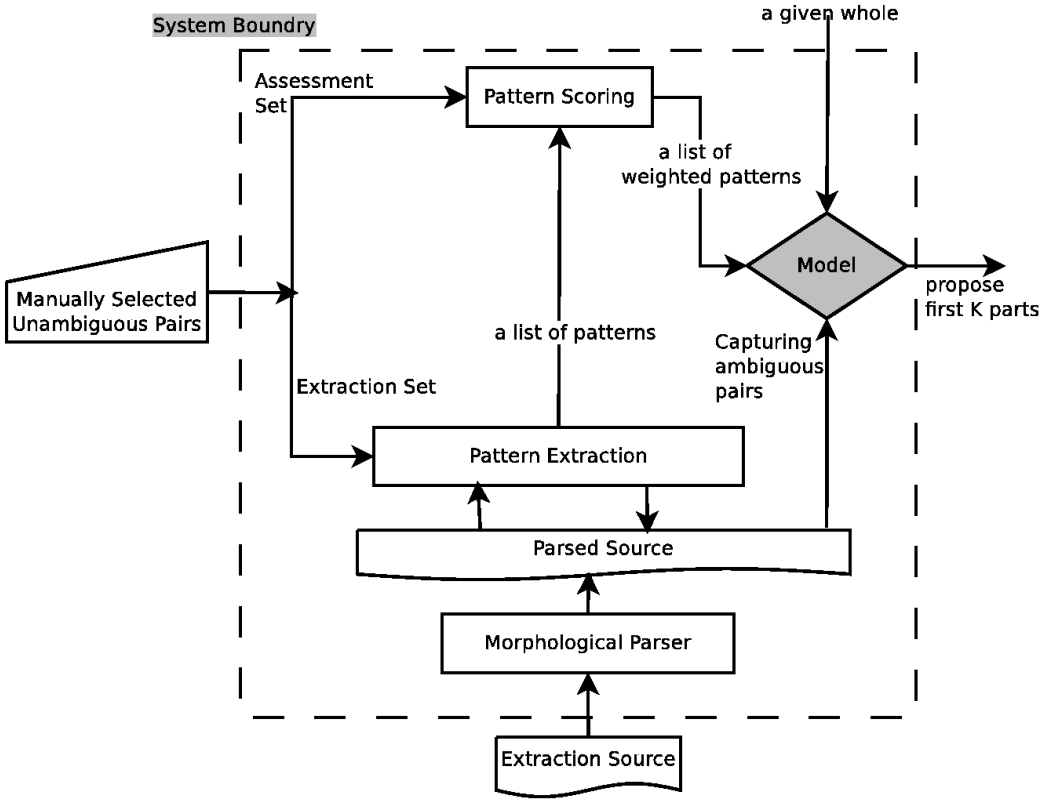

Figure 5.1 High-level Representation of the System

5.6.3.1 Pattern Identification

We began by manually preparing a set of unambiguous seed pairs that convey a part-whole relation. For instance, the pair (engine, car) would be member of that set. The seed set is further divided into two subsets: *an extraction set* and *an assessment set*. Each pair in the extraction set is used as query for retrieving sentences containing that pair. Then we generalized many lexico-syntactic expressions by replacing part and whole token with a wildcard or any meta character. The second set, the assessment set, is then used to compute the usefulness or reliability scores of all the generalized patterns. Those patterns whose reliability scores, *rel(p)*, are very low are eliminated. The remaining patterns are kept, along with their reliability scores. A classic way to estimate *rel(p)* of an extraction pattern is to measure how it correctly identifies the parts of a given whole. The success rate is obtained by dividing the number of correctly extracted pairs by the number of all extracted pairs. The outcome of entire phase is a list of reliable lexico-syntactic expressions along with their reliability scores.

5.6.3.2 Part-Whole Pair Detection

In order to extract the pairs among which there is a part-whole relation, the previously generated patterns are applied to an extraction source that is a Turkish raw text. The instantiated instances (part-whole pairs) are assessed and ranked according to their reliability scores, where reliability score of a pair is described below.

There are several ways to compute a reliability score for both pattern and instance. In [36], the reliability score of a pattern, *rel(p)*, was proposed as shown in equation (5.1) and that of an instance, *rel(i)*, is formulated as in equation (5.2).

$$\text{rel(p)} = \frac{\sum_{p \in P} \frac{pmi(i,p)}{max_{pmi}} \times rel(i)}{|I|} \tag{5.1}$$

$$\text{rel(i)} = \frac{\sum_{i \in I} \frac{pmi(i,p)}{max_{pmi}} \times rel(p)}{|P|} \tag{5.2}$$

pmi is defined in (3.12) at Section 3.4.2.1 as the one of the commonly used metrics for the strength of association between two variables, where max_{pmi} is the maximum pmi value between all pairs and all patterns and where *rel(i)* is the reliability of instance i.

Initially, all reliability scores of instances in set of unambiguous pairs are set to 1. Then, reliability score of a pattern is calculated based on these *rel(i)* scores.

In [36], the pmi score between an instance *i(x,y)* and pattern *p* was formulated as in following equation (5.3).

$$\text{pmi(i,p)} = \log \frac{|x,p,y|}{|x,*,y||*,p,*|} \tag{5.3}$$

where $|x,p,y|$ is the number of times instance *i(x,y)* is instantiated with pattern p, $|x,*,y|$, $|*,p,*|$ are the individual distributions of instance and pattern respectively. However, the defect in the formula is that the *pmi* score always takes negative values. This leads a ranking the reverse of the expected. It must be multiplied by the numbers of all pairs matched by all patterns,$|*,*,*|$. Thus, we redefined the formula as shown in equation (5.4).

$$\text{pmi(i,p)} = \log \frac{|x,p,y||*,*,*|}{|x,*,y||*,p,*|} \tag{5.4}$$

A frequent pair in a particular pattern does not necessarily convey a part-whole relation. Thus, to calculate reliability of a pair, all patterns are taken into consideration as shown in equation (5.2).

In our research, we experimented with three different measures of association (pmi, dice, T-score) to evaluate their performance. All measures explained in Section 3.4.2. We also utilized *inverse document frequency (idf)* to cover more specific parts. The motivation for use of *idf* is to differentiate distinctive features from other common ones.

We categorized our parts into two groups; *distinctive* and *general* parts. If a part of a given whole is inheritable from hypernyms of that whole, we call this kind of part general or inheritable. If, not, we call such part specific or distinctive part. Here, distinctive part means that part of a whole are not hierarchically inherited. E.g. a desk has *has-part* relationship with *drawer* and *segment* as in WordNet. While *drawer* is distinctive part of *desk, segment* is a general part that inherits from its hypernym "*artifact"*. Indeed, it is really difficult to apply this chaining approach to all nouns. Instead of using all hypernym chain, we separated the parts into two basic groups. First, the parts that seems to be general, like, point, side, segment, etc. These can be inherited from upper physical entity. Second, the parts that seems to be distinctive like kitchen of the house.

Thus, to distinguish the distinctiveness, we utilized *idf* that is obtained by dividing the number of times a part occurs in part position by how many pairs retrieved by all the patterns. We observed that the most frequent part instances are top, inside, segment, side, back, front and state, head etc. All of these resemble general features. Evaluation of distinctiveness will be discussed in next sections.

5.6.3.3 Experimental Design

The morphological parser splits a surface token into its morphemes in system architecture as shown in Figure 5.1. The representation of a parsed token is in the form of surface/root/pos/[and all other markers]. When the genitive phrase "*arabanın kapısı*" (door of the car) is given, the parser split it into the parts as below.

English: (door of the car)

Turkish: arabanın kapısı

Structure in Turkish: NPy+nHn NPx+sH

Parsed: arabanın+araba+noun+a3sg+pnon+gen

kapısı+kapı+a3sg+pnon+p3sg

In order to identify lexical forms that express part-whole relations, we manually selected 200 seed pairs. Out of 200 pairs, 50 are used as pattern extraction set to extract the LSPs and 150 are used as assessment set to compute the reliability scores of each pattern, *rel(p).* All sentences containing part and corresponding whole token in extraction set are retrieved. Replacing part/whole token with a meta character, e.g. wildcard, we extracted many patterns.

However, due to the noisy nature of the Web corpus and the difficulties of an agglutinative language, many patterns have poor extraction capacity. Turkish is a relatively free word order language with agglutinating word structures. The noun phrases can easily change their position in a sentence without changing the meaning of the sentence, and only affecting its emphasis. This is a big challenge for syntactic pattern extraction. Based on reliability scores, we decided to filter out some generated patterns and finally obtained six different significant patterns. Here is the list of the patterns, their examples and related regular expression formula:

1. Genitive Pattern: NPy+gen NPx+pos

In Turkish, there is only one genitive form: The modifier morphologically takes a genitive case, *Gen (nHn)* and the head takes possessive agreement *pos(sH)* as shown before ("arabanın kapısı / door of the car"). The morphological feature of genitive is a good indicator to disclose a semantic relation between a head and its modifier. In this case, we found that the genitive has a good indicative capacity, although it can encode various semantic interpretations. Taking the example, Ali's team, and the first interpretation could be that the team belongs to Ali, the second interpretation is that Ali's favorite team or the team he supports. To overcome such problem, researchers have done many studies based on statistical evidence, some well-known semantic similarity measurements and semantic constraints based on world knowledge resources. The regular expression of genitive pattern for "arabanın kapısı" is as follows:

Regex : \w+\+noun[\w\+]+gen

\w+\+noun[\w\+]+p3sg

2. NPy+nom NPx+pos

English: (car door)

Turkish: araba kapısı

Structure in Turkish: NPy NPx+sH

Parsed: araba+araba+noun+a3sg+pnon+nom

kapısı+kapı+noun+a3sg+pnon+p3sg

Regex: \w+\+noun\+a3sg\+pnon\+nom

\w+\+noun\+[\w\+]+p3sg

3. NPy+Gen (NPs|ADJ)+ NPx+Pos

English: (back garden gate of the house)

Turkish: Evin arka bahçe kapısı

Structure in Turkish: NPy+nHn (NPs|Adj)+ NPx+sH

Parsed: Evin+ev+noun+a3sg+pnon+gen

arka+arka+noun+a3sg+pnon+nom

bahçe+bahçe+noun+a3sg+pnon+nom

kapısı+kapı+noun+a3sg+p3sg+nom

Regex: \w+\+noun[\w\+]+gen

(\w+\+noun\+a3sg\+pnon\+nom | \w+\+adj[\w\+]+)

\w+\+noun[\w\+]+p3sg

4. NPx of one-of NPys

English: (the door of one of the houses)

Turkish: Evlerden birinin kapısı

Structure in Turkish: NPy birinin NPx+sH

Parsed: Evlerden+ev+noun+a3pl+pnon+abl

birinin+biri+pron+quant+a3sg+p3sg+gen

kapısı+kapı+noun+a3sg+p3sg+nom

Regex: \w+\+noun\+a3pl\+pnon\+abl

birinin\+biri\+pron\+quant\+a3sg\+p3sg\+gen

\w+\+noun\+\w+\+p3sg

5. NPy whose NPx

English: The house whose door is locked

Turkish: Kapısı kilitli olan ev

Structure in Turkish: NPx+sH (NPs|Adj) olan NPy

Parsed: Kapısı+kapı+noun+a3sg+p3sg+nom

kilitli+kilit+noun+a3sg+pnon+nom-adj*with

olan+ol+verb+pos-adj*prespart

ev+ev+noun+a3sg+pnon+nom

Regex: \w+\+noun[\w\+]+p3sg\+\w+

(\w+\+noun\+a3sg\+pnon\+nom| \w+\+noun\+a3sg\+pnon\+nom\-adj*with)

(\w+\+verb\+pos\-adj*prespart| \w+\+verb\+pos\+narr\+a3sg)

\w+\+noun\+a3sg

6. NPy with NPxs

English: the house with garden and pool

Turkish: bahçeli ve havuzlu ev

Structure in Turkish: (NPx+lH)+ ve? (NPx+lH)? NPy

Parsed: bahçeli+bahçe+adj

ve+ve+conj

havuzlu+havuz+noun+a3sg+pnon+nom-adj*with

ev+ev+noun+a3sg+pnon+nom

Regex: (\w+\+noun\+a3sg\+pnon\+nom\-adj*with)+

(ve+ve+conj)?

(\w+\+noun\+a3sg\+pnon\+nom\-adj*with)?

\w+\+noun\+a3sg

All patterns were evaluated according to their usefulness. To assess them, output of each pattern was checked against a given assessment set. Setting instance reliability of all pairs in the set to 1, reliability score of the patterns are computed as shown (5.1). For a assessment set size of 150 pairs, all pattern and their *rel(p)* are given in Table 5.6.

When comparing the patterns, P1 is the most reliable pattern with respect to all measures. P1 is based on genitive case which many studies utilized it for the problem. We roughly ordered the pattern as P1, P2, P3, P6, P4, and P5 by their normalized average scores in the Table 5.6.

Table 5.6 Reliability of patterns

	rel(P1)	**rel(P2)**	**rel(P3)**	**rel(P4)**	**rel(P5)**	**rel(P6)**
pmi	**1.58**	1.53	0.45	0.04	0.07	0.57
dice	**0.01**	0.003	**0.01**	0.004	0.001	0.003
T-score	**0.11**	**0.12**	0.022	0.0004	0.001	0.03

To calculate reliability of instances, we utilized not only pmi measure, but also dice, T-score and idf measures. In equation (5.1), *rel(p)* and equation (5.2), *rel(i),* association measure can be pmi, pmi-idf, dice, dice-idf, T-score, and T-score-idf. For a particular whole noun, all possible parts instantiated by patterns are selected as a candidate set.

For each association measure, their *rel(p)* and *rel(i)* scores are calculated and further sorted. The first K candidate parts are checked against the expected parts.

5.6.4 Statistical Selection

So far, we have selected first N most frequent parts for a whole by running a given specific pattern from GP or TDK-P. By applying a single pattern to a big corpus, we have taken and evaluated the results. Instead, in this part, we retrieved all candidates part-whole pairs obtained from all patterns (in GP and TDK-P) and built a big whole-by-part matrix, namely global matrix, whose $cell_{ij}$ represents how many times $whole_i$ and $part_j$ co-occurs together, no matter which patterns produce them. In order to compare the clusters GP and TDK-P, we also used two separate bunches, and a big integrated one as well.

The contingency global table, or whole-by-pair matrix, gives us a chance to apply some statistical metrics such as Information Gain (IG), χ 2, etc. If a part particularly occurs with a specific whole, it indicates that there is a meaningful link between them. Or, if a common part mostly appears with many wholes, its global importance is lower than others as formulated in idf. By applying the formulas such as χ 2 value or IG, the global matrix can be converted into scored one, each cell can represent with those scores.

As a baseline algorithm, the number of times a whole and a part co-occurs together can be a reference score. All statistical metrics could be compared with that baseline. Metrics should outperform the baseline algorithm, because they could have expensive computational cost.

5.6.5 Baseline Algorithms

Each approach must have its own baseline algorithm because their circumference can have particular advantage or disadvantage due to many factors. We proposed different baseline formulation for bootstrapped patterns and pre-defined pattern clusters.

To designate a baseline algorithm for bootstrapped patterns, for a given whole, its possible parts are retrieved from a list ranked by association measure between whole and part that are instantiated by a reliable pattern as formulated in equation (5.5).

$$\text{assoc(whole,part)} = \frac{|\text{whole ,pattern ,part}|}{|*,\text{pattern ,part}||\text{who le,pattern ,}*|} \quad (5.5)$$

We intuitively designated a baseline algorithm to compare the results and the expectation is that a proposed model should outperform the baseline algorithm. The baseline function is based on most reliable and productive pattern, the genitive pattern. As Table 5.6 suggests, the rel(genitive-pattern) has the best score in accordance with average of all three measures (pmi, dice and T-score) and the capacity is about 2M part-whole pairs.

For a given whole, all parts that co-occur with that whole in the genitive pattern are extracted. Taking co-occurrence frequency between the whole and part could be misleading due to some nouns frequently placed in part/head position such as side, front, behind, outside. To overcome the problem, the co-occurrence, the individual distributions of both whole and part must be taken into account as shown in equation (5.5). These final scores are ranked and their first K parts are selected as the output of baseline algorithm.

For the evaluation of GP and TDK-P, we applied different baseline algorithm. The matrix shows how many times a given whole and a given part appear together. With this, we can retrieve most frequent N parts for a given whole. This score adds up all frequency number from all patterns, hence, gives a basic line with which we can compare the models.

5.6.6 Challenges

We have faced many issues so far. Here, we have discussed those problems that mostly encountered in this kind of studies alongside with their some solutions.

- Almost all studies suffer from the very basic problem of natural language processing: “ambiguity of sense”. For a given whole, proposed parts could be incorrect due to polysemous words. Girju et al. [119] represented that some of patterns always refer to part-whole relation in English text, while most of them are ambiguous. Their listings of unambiguous and ambiguous patterns are given in Table 5.7. *Part-of* pattern, *genitive construction*, the verb *-to have*, *noun compounds* and *prepositional construction* are classified as ambiguous meronymic expressions. For Turkish domain, we could not easily do such classification and find even one unambiguous pattern to extract part-whole relation. Additional methods are needed to cope with the problem and to find more accurate results from extracted pairs

Table 5.7 Ambiguous and unambiguous pattern list [119]

Unambiguous Patterns	**Ambiguous Patterns**
parts of NPy include NPx	NPx part of NPy
NPy consist of NPx	NPy has NPx
NPy made of NPx	NPy's NPx
NPx member of NPy	NPx of NPy
One of NPy constituents NPx	NPy NPx
	NPy with NPx

- Adoption of the general patterns from other studies to Turkish domain is difficult due to free word order language characteristics of language. The noun phrases can easily change their position in a sentence without changing the meaning of the sentence.
- Determining a window is crucial for the potential parts. Keeping the windows size smaller can lead to losing real parts. However, a larger window leads to many irrelevant NPs extracted with large context and it deteriorates system performance. We observed the window size of 15 allows us to capture more reliable parts and sentences. For example, ultraviyole radyasyon (ultraviolet radiation)-güneş enerjisi(solar energy) is part-whole pair in the following example.

  "Ultraviyole (UV) radyasyon, dünya yüzeyine erişen güneş enerjisinin doğal bir parçasıdır". (Ultraviolet (UV) radiation is a natural part of the solar energy that access to the Earth's surface.)
- The patterns can also encode other semantic relations such as hyponymy or relatedness. Although use of genitive case is very popular for detecting part-whole relations, the characteristic of genitive is ambiguous. The morphological feature of genitive is a good indicator to disclose a semantic relation between a head and its modifier. We found that the genitive has a good indicative capacity as shown in Table 5.6, although it can encode various semantic interpretations. Taking the example, "Ali's team", and the first interpretation could be that the team belongs to Ali, the second interpretation is that Ali's favorite team or the team he supports. It refers such relations "Ali's pencil/Possession", "Ali's father/Kindship", "and Ali's handsomeness/Attribute". Same difficulties are

valid for other patterns. To overcome the problem, statistical evidence has been utilized so far.

- Even the best patterns are not safe enough all the time. The sentence “door is a part of car” strongly represents part-whole relation, whereas “he is part of the game” gives only ambiguous relation. The word “Part of” has nine different meanings in TDK. It means that it is nine times more difficult to disclose the relation. In the following example, corresponding part track belongs to sixth meaning of part-of pattern.

  “albüme adını veren parça, Pink Floyd'un caza yakın parçalarından biridir.”

  (the title track which is the name of albume is one of the parts close to Pink Floyd's jazz).
- Some patterns tend to disclose some particular relations such as Possession, Kindship, Ownership, Attribute, Attachment, and Property which are considered as part-whole relation in this study. Some can retrieve other types of semantic relations such as hyponym, relatedness etc. This will be emphasized in next sections.
- The model mostly needs background knowledge especially for domain specific problem. For instance, when running models on football domain, the model needs an ontology covering facts such as “Manchester United is a football team”.
- Some expressions can be more informal than written language or grammar. Indeed, in any language, different kinds of expression can be appropriate in various situations. From formal to informal, from written to spoken, from jargon to slang, all type of expressions are a part of corpus. This variety can cause another bottleneck for applying regular expression or patterns. For example, following sentence is taken from corpus.

  “Beyinlerimiz televizyonun bir parçasıdır” (Our brain is part of television)
- Some words are not suitable for meronymy relations. Even in WordNet, many synsets have no meronym relation. E.g. how many parts can these words “result” or “point” have? Particularly abstract words are harder than concrete ones in terms of evaluation. Therefore, evaluation must be done depending on word characteristics.

- Rich morphological feature of Turkish language means a barrier for computational linguist to overcome it. It has free word order syntax and complicated morphology. For instance, an English phrase including more than 10 words can be translated into one single Turkish word by means of morphological suffixes.
- Some patterns have very limited capacity. For example, "içeren parçaları" (parts of NPy include NPx) and kısmen (partly) have very poor results. Both are excluded because of the number of returned cases. First pattern returns 2 and latter returns 10 cases only.
- Some wholes have limited parts, for example ithalat (import), baklagiller (legume family), ödeme (payment), başvuru (application), dosya (file), etc.

5.6.7 Results and Evaluation

Three clusters of patterns are taken into consideration. The first two patterns, general patterns and dictionary-based patterns are predefined lists which are obtained from literature and other studies. On the other hand, third cluster of patterns, bootstrapped patterns, are semi-automatically obtained by giving initial unambiguous part-whole pairs.

In evaluation phase, general and dictionary-based patterns are compared to each other due to similar approach and bootstrapped method is analyzed individually. Furthermore, the results pooled from all patterns are evaluated by means of statistical measurements such as, χ 2, IG metrics and etc.

5.6.7.1 Analysis of GP vs. TDK-P

For each category, we selected top 30 words from ranked list and randomly presented them to a user for evaluation. Each category is judged by three people. Rating of user for each word is 0/1 for part-whole relation and 0/1 for strongly associated with category.

Results show precision score of patterns for first 10, 20 and 30 selections in Table 5.8. It indicates that GP are slightly more successful and robust than TDK-P on average. While GP has 64.2%, 61.8% and 56.6% precision, TDK-P has 67.8%, 48.9% and 40.7% for first 10, 20 and 30 parts selection respectively. Moreover, GP are more productive than TDK patterns. The results in Table 5.14 show us production capacity of GP as 12.57 and TDK as 11.91 on average.

At first glance, the most successful results seem to be produced by *with* (from GP), and which *has/have* and *consist-of* (from TDK-P) as shown in Table 5.13. However, evaluation on only precision could be deceptive for some cases. Although we could not measure recall value of the patterns, we considered that evaluation of recall could be discussed over production capacity which is "number of cases per whole whose frequency is bigger than 1", denoted by #ofCpW>1.

The most productive patterns are *has/have* (TDK-P) with production ratio of 42.56. This pattern has also good precision score of 77.8%, hence, a good recall value. *has/have* pattern (GP) has production ratio of 22.6 and its precision is 62.0%. The pattern *member-of* (GP) has ratio of 21.09 and has 53.3% precision value. Recall values can be considered as a function of production capacity where there must be a linear correlation. Thus Table 5.14 suggests that *has/have* pattern (TDK-P) gives promising result.

The highest precision of 81.1% is achieved by pattern *with* (GP). However, it has relatively lower capacity of 11.71 than those patterns discussed. The worst patterns are *made-of* (GP), *made-of* (TDK-P), *constitute-of* (GP) and *family-of* (TDK-P). They have production capacity of 6.5, 6.62, 4.14, and 5.57 and precision rate of 28.3%, 25.7%, 21.7% and 13%, respectively. They showed very poor performance in terms of capacity (recall) and success (precision).

Table 5.8 The precision of GP and TDK-P for the first N selections

GP	**N:10**	**N:20**	**N:30**	**TDK**	**N:10**	**N:20**	**N:30**
part-of	52	52	54	group-of	42	44	41.33
member-of	57.50	53.75	53.33	member-of	80	73	62.67
constitute-of	50	46.25	21.67	amount-of	60	52.50	41.49
consist-of	83.33	80	74.13	family-of	38.18	0	0
made-of	50	52.50	50	made-of	77.18	0	0
has/have	70	67	62	consist-of	97.14	91.35	61.58
with	86.67	80.83	81.11	has/have	80	81.67	77.78
AVG-GP	64.21	61.76	56.6	AVG-TDK-P	67.79	48.9	40.7

5.6.7.2 Analysis of BP

For the evaluation phase, we manually and randomly selected five whole words: book, computer, ship, gun and building. For each whole noun, the experimental results are given in Table 5.9.

Table 5.9 The precision results of the scores for five wholes

Whole	**pmi**	**pmi-idf**	**dice**	**dice-idf**	**T-score**	**T-score-idf**	**Base**	**Avg**
gun-10	2	4	1	1	0	1	2	1.57
gun-20	4	5	3	2	1	1	4	2.86
gun-30	6	6	6	6	2	3	6	5
book-10	9	3	10	10	8	7	8	7.86
book-20	18	9	18	18	16	12	13	14.86
book-30	22	14	22	23	21	20	17	19.86
building-10	4	2	5	7	7	6	7	5.43
building-20	11	8	15	14	15	13	15	13
building-30	17	13	22	23	20	19	18	18.86
ship-10	9	7	9	9	6	5	9	7.71
ship-20	14	13	18	18	9	10	15	13.86
ship-30	18	17	26	24	13	14	21	19
computer-10	8	9	9	9	6	7	8	8
computer-20	16	15	13	15	8	11	10	12.57
computer-30	21	16	20	20	10	15	14	16.57
avg. prec.								
precision10	64	50	**68**	**72**	54	52	**68**	61.14
precision20	63	50	**67**	**67**	49	47	57	57.14
precision30	56	44	**64**	**64**	44	47.3	51	52.86

gun-10 means that we evaluated first 10 selections of all measures for whole gun. For a better evaluation, we selected first 10, 20 and 30 candidates ranked by the association measure defined above. The proposed parts are manually evaluated by looking at their semantic role.

We needed to differentiate part-whole relations from other possible meanings. Indeed, all the proposed parts are somehow strongly associated with corresponding whole. However, our specific goal here is to discover meronymic relationship and, thus we tested our results with respect to the component-integral meronymic relationship as defined in [106] or HAS-PART in WordNet.

Looking at the Table 5.9, for the first 10 selection, all measures perform well against all wholes but gun. This is simply because *gun* gives less corpus evidence to discover parts of it. With a deeper observation, we have manually captured only 9 distinctive parts and 10 general parts, whereas whole *building* has 51 parts, out of which 13 are general parts. For first 10 outputs, *dice-idf* with precision of 72% performs better than others on average. For first 20 selections, *dice* and *dice-idf* share the highest scores of 67%. For first 30 selections, *dice, dice-idf* with precision of 64% outperforms other measures

5.6.7.3 Analysis of Distinctive Parts vs. General Parts

We conducted another experiment to distinguish distinctive parts from general ones. Excluding general parts from the expected list, we re-evaluated the result of the experiments. The results were, of course, less successful but a better fine-grained model is obtained. The result is shown in Table 5.10. The table shows that all idf weighted measures are better than others. For the first 30 selection, when idf is applied, *pmi*, *dice* measures are increased by 2% and *T-score* measure is increased by 7.3% on average as expected.

General parts can easily capture when running the system for "*entity"* or any hypernym. To do so, we checked noun "şey" (thing) to cover more general parts or features. We retrieved some meaningful nouns such as top, end, side, base, front, inside, back, out as well as other meaningless parts.

As a result, we evaluated the distinction problem through bootstrapped pattern due to its production capacity, simplicity and quick evaluation. Similar results can be obtained through other predefined patterns as well. Table 5.10 shows performance of pmi, dice, T-score and their idf weighted counterparts and baseline metrics in terms of distinctiveness. There are two clear observations here:

1. Idf weighted metrics are better than others as expected. Idf eventually can discriminate particular parts by definition, because low-frequent terms have higher idf value. Thus they can represent distinctive part.
2. Dice-based formulas outperform other two metrics, pmi and T-score. Table 5.8 also indicates that only metric which can surpass baseline algorithm is dice and its idf counterpart.

Additionally, we can easily apply IS-A relation, whereas we cannot always apply the same principle to part-whole hierarchy. For instance if tail is a meronym of cat and tiger is a hyponym of cat, by inheritance, tail must be a meronym of tiger then. However, transitivity could be limited in the part-whole relation. Handle is meronym of door; door is a meronym of house. It can incorrectly imply that the house has a handle. On the other hand, finger-hand-body hierarchy is a workable example to say that a body has a finger.

Table 5.10 The results for distinctive parts precision

	pmi	**pmi-idf**	**dice**	**dice-idf**	**T-score**	**T-score-idf**	**Baseline**	**Avg**
prec10	50	50	58	**64**	34	44	60	51.43
prec20	48	50	48	**53**	34	40	51	46.29
prec30	40.67	42.67	47.33	**49.33**	31.33	38.67	40.67	41.52

5.6.8 Statistical Measurements

Table 5.11 shows the performance of a list of statistical metrics on whole-by-part global data. The resulting table has three bunches, first gives the results for data obtained from GP, second is regarding TDK-P and third one is an integrated bunches. Under each bunch, scores from IG, X^2, T-score, dice, Frequency (baseline) approach are represented. For the bunch GP, the ranking is T-score > IG > dice > Freq > X^2. For TDK-P it is T-score > dice > IG > Freq > X^2, which is akin to GP, where IG and dice are swapped.

As another result for BP, Table 5.12 partly confirms our expectation that the success rate from a larger training seed set is slightly better than those from a smaller one. As we increased the seed size from 50 to 150, only *pmi* measure clearly improved and the other measures did not show significant improvements.

Table 5.11 Statistical measurements for GP vs. TDK-P

Patterns	**SM**	**10**	**20**	**30**
GP	IG	66.7%	65.0	58.9
	X^2	44.4	36.1	35.6
	T-score	**74.4**	**70.6**	**66.3**
	dice	66.7	59.4	54.8
	Freq	48.9	43.3	41.5
TDK-P	IG	70.0	62.8	58.1
	X^2	55.6	45.0	43.0
	T-score	**72.2**	**68.3**	**61.5**
	dice	70.0	65.6	61.1
	Freq	63.3	57.8	55.9
AVG	AVG-IG	68.3	63.9	58.5
	AVG-X^2	50.0	40.6	39.3
	AVG-T-score	**73.3**	**69.4**	**63.9**
	AVG-dice	68.3	62.5	58.0
	AVG-Freq	56.1	50.6	48.7

Another important observation is that T-score value shows the best performance. However, X^2 does not even outperform the baseline algorithm within each bunch. T-score, IG and dice formula are the most successful metrics.

Table 5.12 The precision (prec) results for training set (TS) size of 50,100 and 150

#of_TS	results	pmi	pmi-idf	dice	dice-idf	T-score	T-score-idf	Base	Avg
train50	prec10	64	52	70	68	52	44	68	59.71
	prec20	59	50	70	68	48	45	57	56.71
	prec30	51.33	43.33	62.67	**64**	42	46	50.67	51.43
train100	prec10	68	50	72	70	52	48	66	60.86
	prec20	63	49	68	66	48	44	58	56.57
	prec30	56	44.67	63.33	**64**	42.67	45.33	50.67	52.38
train150	prec10	64	50	68	72	54	52	68	61.14
	prec20	63	50	67	67	49	47	57	57.14
	prec30	56	44	**64**	**64**	44	47.33	50.67	52.86

Main advantage of statistical selection is to integrate all results coming from heterogeneous patterns, where each pattern has different success rate, production capacity, tendency to meronymy subtype, e.g. attachment, possession. Merging all output from all patterns can increase recall value of the model and cover many wholes in a broader scope because each single pattern can have its own potential whole and tendency. Some could not take the whole as a parameter. We evaluated those predefined patterns on whole terms which are already produced in advance. Therefore, the difference in success ratio between the patterns could be compared in terms of various aspects. Looking at the Table 5.11, the model proposed here gives a promising result in terms of precision.

Table 5.13 Best patterns of GP vs. TDK-P

#ofParts	GP with	TDK consist-of	TDK has/have	GP T	TDK T	Bootstrap dice-idf
10	86.67	97.14	80.0	74.4	72.2	72
20	80.83	91.35	81.7	70.6	68.3	67
30	81.11	61.58	77.8	66.3	61.5	64

5.6.9 Production Capacity and Recall Estimation

Table 5.14 shows number of cases, number of wholes proposed by each pattern and their success rates in precision. We also selected those wholes whose frequency is higher than 1 to decrease error rate coming from false matching. At first glance, the most successful pattern is with (GP) when ranking them according to precision for first

30 selections. Production capacity denoted by #ofCpW>1 and success ratio can be combined to evaluate them within different aspects, where production capacity does not refer to how many cases matched by corresponding pattern, but how many cases matched per whole on average. By multiplying the success rate and the normalized value of #ofCpW>1 (number of cases for per whole whose frequency is bigger than 1), we get another ranking factor representing and combining both precision and production capacity, we got following priority of patterns; has/have (TDK-P), has/have (GP), member-of (TDK-P), with (GP), part-of (GP). The pattern has/have (TDK-P) has both good production rate of 42.59 and precision rate of 77.8% therefore, it appears in first place in combined ranking. The poorest patterns are family-of and amount-of (TDK-P) according to new ranking factor.

Table 5.14 Ranked by success rate in precision of each pattern

Cluster	**P**	**#ofC**	**#ofW**	**#ofW>1**	**#ofC>1**	**#ofCpW>1**	**S30**
GP	with	68.8K	8.7K	5.6K	65.7K	11.71	81.1
TDK-P	has/have	445K	13.7K	10.3K	442K	42.56	77.8
GP	consist-of	9.2K	2K	1K	8.2K	8.13	74.1
TDK-P	member-of	20K	3.9K	2K	18.2K	8.62	62.7
GP	has/have	12K	8.2K	5.1K	117K	22.66	62
TDK-P	consist-of	12.4K	2.7K	1.4K	11K	7.79	61.6
GP	part-of	19.3K	2.4K	1.3K	18.1K	13.75	54
GP	member-of	23K	2K	1K	22K	21.09	53.3
TDK-P	amount-of	3.4K	1.4K	5.5K	7.5K	1.37	41.5
TDK-P	group-of	22.7K	3.6K	2K	21K	10.85	41.3
GP	made-of	6.3K	1.7K	836	5.4K	6.50	28.3
TDK-P	made-of	4.9K	1.4K	612	4K	6.62	25.7
GP	constitute-of	598	293	97	402	4.14	21.7
TDK-P	family-of	184	47	30	167	5.57	13

Another smooth evaluation might be done over correlation between success (precision) and some factors such as number of cases, wholes, cases per whole and others. When looking at correlation Table 5.15, the success of a pattern mostly and strongly depends on number of producing unique wholes.

Second is #ofW>1 and #ofCpW>1. This finding is worth to discuss more deeply. The number of cases matched by a given pattern has secondary importance. The essential point is how many unique wholes and number of cases per each whole a pattern can extract. As shown in Table 5.14, for some patterns, e.g. made-of (GP, TDK-P) and amount-of (TDK-P), although they have a good capacity for matching cases, but they

have poor #ofCpW>1 score. Thus, this kind of scattered patterns does not show a signifcant performance.

In Table 5.14, P is patterns, #ofC is number of cases, #W is number of whole, #ofW>1 is number of whole whose frequency is greater than 1, #ofC>1 is number of cases whose whole are seen more than 1 times, #ofCpW>1 is number of cases per whole whose frequency is greater than 1, S30 is success rate for 30 candidates.

Table 5.15 Correlation table

Correlation	**SuccessRate**
#ofCases	0.492
#ofWhole	0.7144112
#ofW>1	0.6054463
#ofC>1	0.4874499
#ofCpW>1	0.545055

5.6.10 Semantic Relatedness

The goal of this section is to retrieve meronymic relations. However we had a diversity of patterns and they can disclose other semantic relations. Then it opens another case on considering semantic relatedness when evaluating the results. Semantic relatedness is defined as "how much two concepts are semantically distant or close in a network or taxonomy by using all relations between them (i.e.hyponymic/ hypernymic, antonymic, meronymic and any kind of functional relations including is-made-of, is-an-attribute-of, etc.)" [40].

It is analyzed that candidate parts fall into many semantic relations (SR) such as attachment, attribute, ownership, hyponym, synonym etc. While some are taken into account as part-whole relation as shown in Table 5.16, some are considered as others. Table 5.16 includes three columns; the semantic relations that we have encountered and labeled as a part-whole relation, patterns that are able to explore those relations and some instances seen in corpus. A semantic relation can be, of course, disclosed more than one pattern. However we demonstrated them with only one pattern in Table 5.16. Each pattern might have its own tendency to some specific semantic relations. On the other hand, if we evaluate the performance of the models in terms of broader semantic relatedness, then we can obtain better results as expected. For example, there is a hypernym relationship between whole:futbol-spor (football-sport) or there is a synonym relationship between whole:yaşam-hayat (life).

Table 5.16 SR with examples interpreted in corpus and patterns for GP vs. TDK-P

Semantic Relations	**Pattern**	**Examples**
Possession/Ownership	has/have	government:bank
Kindship	with	woman:kid
Property/Attribute	with-has/have	person:talent
Attachment	with	woman:skirt
Location	part-of	World:Europe
Purpose	consist-of	team:achievement
Topic	consist-of	art:exhibition
Measure	amount-of	Turkey:population
Make/Produce	made-of	television:broadcast

All the results of pattern clusters are checked against semantic relatedness. Table 5.17 covers the scores of each pattern in GP and TDK-P for 10, 20 and 30 candidates. All patterns in GP are increased by score 9.5%, 17% and 11.5% for first 10, 20 and 30 selections, respectively. Similar results are observed in TDK-P except for made-of pattern. There is no change in score for made-of pattern because of the small number of cases. The average score for first 10, 20 and 30 selections are increased 11.31%, 9% and 11.03%, respectively.

Table 5.17 Comparison of precisions of GP and TDK-P for semantic relatedness

GP+Related	**10**	**20**	**30**	**TDK-P+Related**	**10**	**20**	**30**
part-of	56.0	56.0	58.7	group-of	68.0	68.0	66.0
member-of	87.5	77.5	75.0	member-of	96.0	89.0	84.0
constitute-of	67.5	72.5	31.7	amount-of	65.0	55.0	45.7
consist-of	91.7	90.0	86.2	family-of	67.0	0.0	0.0
made-of	85.0	77.5	71.0	made-of	77.18	0.0	0.0
has/have	72.0	69.0	67.3	consist-of	98.6	92.1	63.8
with	91.7	88.3	86.7	has/have	81.7	83.3	80.6
AVG-GP	78.8	75.8	68.1	AVG-TDK	79.1	77.5	68.0

When we evaluated the semantic relatedness on each statistical result, the results of bootstrapped patterns with respect to these SR that includes part-whole as well, we obtained better precision. Looking at the result in Table 5.18, the average score for first 10, 20, 30 selections is 68%, 67%, 64% and 90%, 84%, 79,3% with respect to part-whole and SR, respectively. Similar performance is observed in terms of average score as shown in Table 5.19. There is an approximately 6% improvement on average for all 10, 20 and 30 candidates. Here, statistical approach slightly outperforms other methodologies.

Table 5.18 The results of BP for part-whole relation and SRs

	Part-Whole			**SRs**		
whole	**10**	**20**	**30**	**10**	**20**	**30**
Gun	1	3	6	5	11	14
Building	5	15	22	101	19	28
Computer	9	13	20	10	16	24
Ship	9	18	26	10	20	28
Book	10	18	22	10	18	25
Average	68	67	64	90	84	79.3

Table 5.19 Performance in precision of statistical metrics for semantic relatedness

Patterns	**SM**	**10**	**20**	**30**
GP-Related				
	IG	68.9	68.9	65.2
	X^2	51.1	45.0	44.1
	T-score	76.7	75.6	71.5
	dice	68.9	64.4	60.4
	Freq	52.2	49.4	47.8
TDK-Related				
	IG	76.7	68.9	64.8
	X^2	62.2	50.0	44.1
	T-score	78.9	76.1	69.6
	dice	76.7	73.3	68.5
	Freq	71.1	66.7	63.0
AVG-Related				
	AVG-IG	72.8	68.9	65.0
	AVG-X^2	56.7	47.5	44.1
	AVG-T-score	77.8	75.8	70.6
	AVG-dice	72.8	68.9	64.4
	AVG-Freq	61.7	58.1	55.4

CHAPTER 6

SYNONYM

The automatic extraction of synonym is a major task in NLP. Synonym is still a serious topic of debate between strict and traditional definitions. One is described as “words identical in their meaning" and latter is the broader ones which covers near-synonym; expressions are so similar but not identical. Synonym is defined as “Expressions with the same meaning are synonymous” to introduce a notion of absolute synonym [1]. Expressions are absolute synonym; if all their meanings are identical, they are synonymous in all contexts and they are semantically equivalent in all the dimensions of meaning [2]. When considering the all the condition, absolute synonym is very rare and capturing is very hard.

For this reason, recent studies introduced the synonym definition as words or phrases with similar or identical meanings. Basic idea behind this is that semantic similarity is sufficient to detect synonym. Proposed model for this approach is called as distributional similarity which has been widely used to capture the semantic relatedness of words. The underlying assumption of this approach is distributional hypothesis that semantically similar words share similar contexts. Distributional similarity of words sharing a large number of contexts could be informative [56].

Following this idea, various studies have been proposed for automatic synonym acquisition. Recent studies were generally based on distributional similarity. However, this methodology itself can be ambiguous and insufficient. Because distributional similarity approach can cover other semantically related words and might not distinguish between synonyms and other relations. For example, list of top-10 distributionally similar words for orange is: yellow, lemon, peach, pink, lime, purple, tomato, onion, mango, lavender [134].

In addition, pattern-based approach can be auxilary for synonym extraction. It is the most precise acquisition methodology earlier applied by Hearst [5] and relies on LSPs. The pattern-based approach tends to capture hyponymy and meronymy relations as well, whereas it is aparently incompatible and insufficient for synonym detection. Thus, pattern-based approach or external features such as grammatical relations can be integrated into distributional similarity approach for identifying synonyms by narrowing distributional context. Although some studies have showed that classical distributional methods always have a higher recall than pattern based techniques in this area [135], integrating two or more approaches were reported that system performance was improved [133], [135], [136], [137].

The identification of synonym relation from text helps to address various NLP applications, such as information retrieval and question answering [39], [124], [125], [126], [127], [128], automatic thesaurus construction [7], [129], automatic text summarization [130], language generation [131], English lexical substitution task [132], lexical entailment acquisition [133].

In this study, two models are proposed: one is for detecting synonymy and latter one is for extracting synonym. Objective of Model-1 is to determine synonym nouns in a Turkish corpus by relying on distributional similarity that is based on syntactic features (obtained by dependency relations) and semantic features obtained by syntactic patterns and LSPs respectively. The features of the proposed model consist of co-occurrence statistics, four semantic relations and ten syntactic dependency relations where a pair of words are represented with fifteen different features and a target class (SYN/NONSYN).

In Model-2, overall objective is to determine synonym nouns in a monolingual Turkish Corpus by relying on distributional similarity and the dictionary definitions. The model is designed so that for a target word, it automatically proposes a list of candidate words and determines whether there is synonymy or not between two words. For each target word, the candidate lists are built by means of dependency relations. The closest K words for a given target word are taken as candidate list. The similarities between target and each candidate are computed by many different features with a large variety of measurements as explained in latter sections. Each pair is represented in terms of those features that are from distributional characteristics to dictionary definitions. Thanks to on-line bilingual dictionary (Turkish-English), WordNet Similarity packages [20] are

utilized as well. Class labels of the pairs are obtained from monolingual on-line dictionaries. Finally, machine learning algorithms are successfully applied on the data including all these useful extracted features.

6.1 Related Works

As one of the most well-known semantic relations, synonymy has been subject to numerous studies and a variety of methods have been proposed to automatically or semi-automatically detect synonyms from text source, dictionaries, wikipedia, search engines. Among them, the most popular methods are based on distributional hypothesis [56] which adopts that semantically similar words share similar contexts. The process of this approach is as follows: co-occurrence, syntactic information, dependency relations, etc. of the words surrounding the target word are extracted as a first step. Afterwards target word is represented as a vector with these contextual features. At the second step, the semantic similarity of two terms is evaluated by applying a similarity measure between their vectors. The words can be ranked according to their scores. Finally, top candidates are selected as most similar words from ranked list.

There have been many studies [7], [138], [139], [140], [141] which used distributional similarity to the automatic extraction of semantically related words from large corpora. Distributional approaches are applied into monolingual corpora [7], [142], [143], monolingual parallel corpora [144], [145], bilingual corpora [144], [146], multilingual parallel corpora [147] and monolingual [148], [149], bilingual dictionaries [134]. Some of the studies [56], [150], [151], [152], [153], [154], [155], [156] are relied on multiple-choice synonym questions such as SAT analogy questions, TOEFL synonym questions, ESL synonym-antonym questions. These studies can vary with respect to usage of weighting scheme, similarity measurement, grammatical relations etc.

However most of these studies are not individually sufficient for synonym. Because the approach also covers near-synonyms and does not distinguish between synonyms and other relations. Hence, recent studies used different strategies: integrating two independent approaches such as distributional similarity and pattern-based approaches, utilizing external features or ensemble method with combining the results to obtain more accuracy. Mirkin [133] integrated pattern-based and distributional similarity methods to acquire lexical entailment. Firstly, they extracted candidate entailment pairs for the input term by these methods.

Another study [158] emphasized that selection of useful contextual was important for the performance of synonym acquisition. So that they extracted three kinds of word relationships from corpora: dependency, sentence co-occurrence, and proximity. They utilized VSM, tf-idf weighting scheme and cosine similarity. Dependency and proximity perform relatively well by themselves. The performance of combination of all contextual information gave the best result. Other study of Hagiwara [136] proposed a synonym extraction method with using supervised learning based on distributional and/or pattern-based features. They constructed five synonym classifiers: Distributional Similarity (DSIM), Distributional Features (DFEAT), Pattern-based Features (PAT), Distributional Similarity and Pattern-based Features (DSIM-PAT) and Distributional and Pattern-based Features (DFEAT-PAT). When the comparison was done, the performance of DFEAT over DSIM glittered in all the evalutaion results. The result of combination DFEAT-PAT showed that it was necessary effort to combine them.

Other study [137] used three vector-based models to detect semantically related nouns in Dutch. They analyzed the impact of three linguistic properties of the nouns. They compared results from a dependency-based model with context feature with 1st and 2nd order bag-of-words model. They examined the effect of the nouns' frequency, semantic specificity and semantic class.

As one of the recent studies, [157], graded relevance ranking problem was applied to discover and rank quality of the target term's potential synonyms. The method used supervised learning method; linear regression with three contextual features and one string similarity feature. The method was compared with two different methods [136], [154] and proposed method outperformed the existing ones.

In this study, we designed two models: one is based on only corpus-based features to determine synonymy and other relies on features from WordNet and monolingual on-line dictionary definitions besides corpus-based features for extracting synonymy. In Model-1, our main assumption is that synonym pairs show similar semantic and dependency relation by the definition. They share same meronym/holonym and hypernym/hyponym relations. Contrary to synonymy, hypernymy and meronymy relations can probably be acquired by applying LSPs to a big corpus. Such acquisition might be utilized and ease detection of synonymy. Likewise, we utilized some particular dependency relations such as object/subject of a verb etc. Machine learning algorithms are applied on all these acquired features. The first aim is to find out which dependency

and semantic features are the most informative and contribute most to the model. Performance of each feature is individually evaluated with cross validation. The model that combines all features shows promising results and successfully detects synonymy relation.

In Model-2, beside the usage of features from corpus such as dependency, semantic and co-occurrence, we integrated the power of WordNet and monolingual on-line Turkish dictionary definition. Within this framework, this study is considered being a major attempt for extracting synonyms in Turkish based on corpus-driven distributional similarity approach with combining different features that are obtained by dependency relations, semantic relations, monolingual Turkish dictionaries, bilingual on-line dictionary (Turkish-English) and WordNet.

6.2 Methodology

In this study, we designed two models: Model-1 and Model2. Model-1 only depends on corpus-based features such as co-occurrence, depenency relations and semantic relations. On the other hand, Model-2 uses combination of other features such as WordNet and monolingual on-line Turkish dictionary definitions besides corpus-based features.

In both model, in order to compute the similarity between concepts and eliminate incorrect candidates, we used the cosine similarity measurement based on the word space model which is a representational Vector Space. Vector representation of words gives strong distributional indication for synonymy detection.

Similarity measurement between two vectors sometimes needs term weighting. Weighting scheme for context vectors might be normalization, pmi, dice, jaccard or raw frequency. The scheme can vary depending on the problem; therefore, it must be tested on the domain. Since we do not observe any significant improvements between the weighting formulas, raw frequency is used for context vectors.

6.2.1 Model-1: Synonym Detection

A good way to evaluate system performance is to compare the results to a gold standard. First, as gold standard, human judgments about the similarity of pairs of word are used. We manually and randomly selected 200 synonym pairs and 200 non-

synonym pairs to build a training data set. Secondly, non-synonym pairs are especially selected from associated (relevant) pairs such as tree-leaf, student-school, computer-game etc. Otherwise, selection of irrelevant pairs for negative examples can lead to false induction. The model is considered accurate if it can distinguish correct synonym pairs from relevant or strongly associated ones.

6.2.1.1 Features from Corpus

Our methodology in Model-1 relies on the assumption that synonym pairs mostly show similar dependency and semantic characteristics in corpus. They share the same meronym/holonym relations, same particular list of governing verbs, adjective modification profile and so on, by definition. Even though it is no-use applying LSPs to extract synonymy, acquisition of other semantic relations such as meronymy could be easily done by simple string matching utilization and morphological analysis. By means of the acquisitions, the proposed model can determine if a given word pair is synonym or not. All attributes are based on relation measurements between pairs. For each synonym pair, 15 different features were extracted from different models: co-occurrence (1), semantic relations based on LSPs (4) and grammatical relations based on syntactic patterns and head-modifier relation (10).

Co-occurrence The first feature was gathered statistics about the co-ocurrence of word pairs withing a broad context (window size is equal to 8 from left and right) from corpora. Contrary to hypernymy and meronymy relation, it is seems impossible to directly extract synonym pairs by applying LSPs to a big corpus. Synonym pairs are not likely to co-occur together in same context and specific patterns at the same time. Therefore, first-order distributional similarity does not work for synonyms. At least, second order representation is needed. Simple co-occurrence measure might not be used for synonymy but non-synonymy. Their co-occurrence could be lower than relevant pairs. We experimentally selected dice metric to measure co-occurring feature. It is computed by roughly dividing the number of co-occurrence by summation of marginal frequencies of words.

Meronym/Holonym: For the relation, three different clusters of LSPs are analyzed in Turkish corpus; General (GP), Dictionary-based (TDK-P) and Bootstrapped patterns (BP). Details of extraction process of each cluster are given in Chapter 5. Summary of patterns is listed in Table 6.1. Meronymy/holonymy is used to detect synonymy

relation. After applying LSPs, some elimination assumption and measurement metrics such as X^2 or pmi to acquire meronym/holonym relation, we obtained a big matrix in which rows depict whole candidates, columns depict part candidates and cells represent the possibility of that corresponding whole and part are in meronymy relation. To measure the similarity of meronymy profile of two given words, cosine function is applied on two rows indexed by two given words. Applying cosine function on two columns gives the similarity of holonym profile.

Table 6.1 General Patterns, Dictionary-based Patterns and Bootstrapped Patterns

GP	**TDK-P**	**BP**
NPx part of NPy	Group-of (whole|group|all|set|flock|union)	NPy-gen NPx-pos
NPx member of NPy	Member-of (class|member|team) (from the family of Y)	NPy-nom NPx-pos
NPy constituted of NPx	Amount-of (amount|measure|unit)	NPy-Gen (N|ADJ)+NPx-Pos
NPy made of NPx	Has/Have (Y has l(H))	NPy of one-of NPx
NPy consist of NPx	Consist-of	NPx whose NPy
NPy has/have NPy	Made-of	NPxs with NPy
NPy with NPx		

Hyponym/Hypernym: Same procedure in meronymy acquisition holds true for hypernymy and hyponymy relation. One relation matrix is built for hyponymy/hypernymy by applying LSPs and same procedure is carried out. The most important LSPs for Turkish that are given in Chapter 4, are as follows:

1. “NPs gibi CLASS” (CLASS such as NPs),

2. “NPs ve diğer CLASS” (NPs and other CLASS)

3. “CLASS lArdAn NPs” (NPs from CLASS)

4. “NPs ve benzeri CLASS” (NPs and similar CLASS)

First pattern gives strong indication of IS-A hierarchy. Given the syntactic patterns above, the algorithm extracts the candidate list of hyponyms for a hypernym.

Dependency Relations: The dependency relations are obtained by syntactic patterns (or regular expression). For example, for auto and car pair, possible governing verbs bearing direct-object relations might be drive, design, produce, use etc. The dimension of word-space model of direct-object syntactic relation consists of verbs and the cells indicate the number of times the selected noun is governed by corresponding verb. The

more they are governed by the similar verb profile, the more likely they are synonyms. Likewise, the process is naturally applicable for other syntactic features. The more they are modified by same adjectives, the more likely they are synonym. Although 36 different patterns were extracted, 8 were eliminated because of the poor results. Then we grouped them according to their syntactic structures. Representation of groups, number of patterns and examples in English-Turkish were given in Table 6.2.

Table 6.2 Dependency features

Features	**Dependency Relation**	**# of Patterns**	**Examples**
G1	direct object of verb	13	I drive a car Araba sürüyorum
G2	subject of verb	3	Waiting car Bekleyen araba
G3	direct object/subject of verb	3	-
G4	modified by adjective+(with/without)	2	Car with gasoline Benzinli araba
G5	modified by inf	1	Swimming pool Yüzme havuzu
G6	modified by noun	1	Toy car Oyuncak araba
G7	modified by adjective	1	Red car Kırmızı araba
G8	modified by acronym location	1	The cars in ABD ABD'deki arabalar
G9	modified by proper noun location	1	The cars in Istanbul İstanbul'daki arabalar
G10	modified by locations	2	The car at parking lot Otoparktaki araba

The essential problem we faced in the experiments is the lack of features of some words. Particularly, rare words cannot be represented due to lack of corpus evidence. Even in the corpus that contains about 500M words, all instances of use of Turkish language may not be present. Thus, those instances in train data that do not occur in any of dependency and semantic relations are eliminated. Especially the pairs including low frequent word cannot be represented and evaluated by means of the methodology as the number of missing values in many features increases. Out of 400 instances, about 40-50 were discarded from training data due to insuffciency.

Synonym Classification for Model-1: Finally, train data turns out to contain balanced number of negative and positive examples with fifteen attributes. All the cells contain real value between 0-1. We know and accept that all features but co-occurrence feature have positive linear relationship with target class. Therefore, the data is considered to

exhibit linear dependency. As a consequence of linearity, linear regression is an excellent and simple approach for such a classification. It has been widely used in statistical applications. The most suitable algorithm is logistic regression which can easily be used for binary classification in the domains with numeric attributes and nominal target class. Contrary to the linear regression, it builds a linear model based on a transformed target variable.

6.2.1.3 Results and Evaluation of Model-1

To evaluate the impact of semantic and dependency relations in detecting synonyms, first, we looked at their individual performances in terms of cross-validation. Picking up each feature one by one with target class, we evaluated the performance of logistic regression on the projected data. As long as the averaged F-measured score of the corresponding feature is higher than 50%, it is considered a useful feature otherwise, independent feature.

The first aim is to find out which feature is the most informative for detecting synonymy and contributes most to the overall success of the model. When evaluating the result as shown in Table 6.3, the semantic features are notably better than syntactic dependency models in finding true synonyms. They are called to be good indicators.

Table 6.3 F-Measure of semantic relations (SRs) features

	Co-occurrence	**Hyponym**	**Hypernym**	**Meronym**	**Holonym**
F-Measure	62.5	60.5	60	**68.7**	**73.7**

Among semantic relations, the most powerful attributes are meronymy and holonymy features with F-measure of 68.7% and 73.7%, respectively. The possible reason for the success seems to be the suffcient number of cases matched by lexico-syntactic and syntactic pattern from which semantic and syntactic features are constructed. For example, the model utilizing meronymy relations has a good production capacity and success. The Table 6.4 shows that meronymy-holonymy matrix has the size of 17K x 18K. The total number of instance is 1.7M. Mero is Meronym, Hypo is Hyponym, AVG_cpr is average case per row and AVG_cpc is average case per column in Table 6.4.

Average number of instances for each meronym is 102 and for each holonym are 96. They also show good performance. The averaged number of instances for hypernymy

and hyponymy are 50 and 8, respectively. As a result of insuffcient data volume, hypernymy/hyponymy semantic relation is relatively weaker than meronymy.

Table 6.4 Statistics for features

	#ofrow	**#ofcol**	**#ofcases**	**AVG_crp**	**AVG_cpc**
G1	16K	1.7K	3.3M	206	2010
G2	18K	1.7K	3M	164	1783
G3	10K	1.4K	0.5M	47	341
G4	13K	5K	1.6M	128	319
G5	7K	1.6K	1M	140	621
G6	13K	13K	5.3M	391	405
G7	20K	5.6K	12M	590	2106
G8	6K	1.6K	0.1M	23	86
G9	1.7K	0.2K	0.01M	7	51
G10	13K	5K	1M	75	195
Mero	17K	18K	1.7M	102	96
Hypo	4.3K	29K	0.2M	50	8

Among dependency relations, G1, G4 and G7 have better performance as shown in the Table 6.5. Also their production capacities are suffcient as well. The poorest groups, G8 and G9, have low production capacity and their performances are worse. As a consequence of the poor results, they are called independent and useless variables. Co-occurrence feature has negative linear relation with target class and its individual performance is 62.5%. It is acceptable as a useful feature.

Table 6.5 F-measure of dependency relations features

	G1	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**	**G8**	**G9**	**G10**
F-measure	**64.7**	58	60.5	**65**	61.6	58.8	**63**	49.4	48.3	62.6

The successful features are linearly dependent on target class. The most suitable machine learning algorithm is the logistic regression. After aggregating all useful features which have better than the individual performances, the machine learning process is carried out and evaluated. The achievement of aggregated model is evaluated in terms of cross validation. On the aggregated data where all useful features are considered, the performance of logistic regression is F-measure of 80.3%. The achieved score is better than the individual performance of each feature. The number of useful features is obviously the main factor to get higher scores. The proposed model utilizes only a huge corpus and morphological analyzer and it receives an acceptable score.

6.2.2 Model-2: Synonym Extraction

The methodology employed here is to identify the synonym pairs from a huge Monolingual Turkish corpus of 500M tokens, an on-line bilingual dictionary, two monolingual dictionaries and WordNet. A Turkish morphological parser is used to parse the corpus [44]. We manually and randomly selected 250 target words for the test of the synonymy extraction model. To improve the system performance, a variety similarity measurement was applied. The list of features extracted from all these resources are generally explained in the following section. Figure 6.1 represents the model architecture and the derived features within a rough view.

6.2.2.1 Features

In this study, word pairs (target/candidate words) are represented by a set of features compatible with machine learning algorithms. Features are extracted from three different resources: monolingual corpus, WordNet and monolingual on-line dictionary. For class label, monolingual on-line synonym dictionaries are used to tag a given synonym pair.

Features from monolingual corpus: Our corpus-based feature extraction methodology relies on the assumption that synonym pairs mostly show similar dependency and semantic characteristics in monolingual corpus. In terms of semantic relation, if words share same meronym/holonym relations and hyponym/hypernym, they are more likely synonymous. With a similar approach, they can have same particular list of governing verbs, adjective medication profile and so on. Fifteen different features are extracted: co-occurrence, semantic relations based on LSPs and dependency relations based on syntactic patterns and head-modifier relation as described in section 6.2.1.1.

All the features that are obtained by corpus are used in Model-2, except G8 and G9. Because they are the poorest groups. So only the name of G10 is labelled as G8 in Model-2.

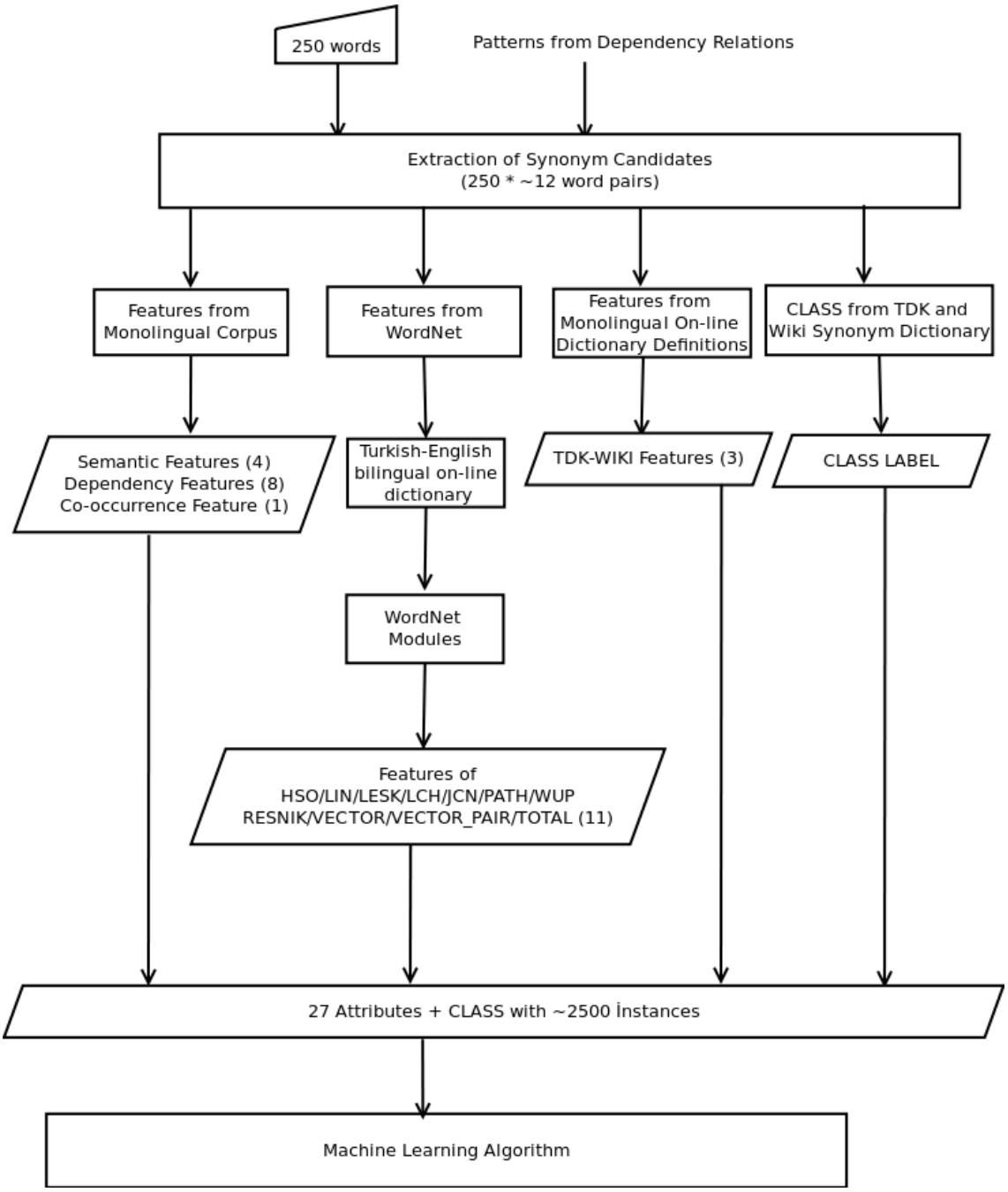

Figure 6.1 Representation of the Synonym Extraction in Model-2

Candidate Synonym Selection: To harvest the candidates of a given target word, system only incorporate with the dependency relations because of production capacity and its simplicity. The time cost is prominently cheaper than others. We looked at which dependency groups are strong indicators or predictors to which semantic relations. 250 words are used to disclose tendency between dependency groups and semantic relation. The most informative dependency features are determined to initiate a candidate list for each word.

For this purpose, the first K, which is chosen 6, nearest neighbors' words to a given target word is selected in terms of dependency features. For each target word and its nearest words, four semantic relations are considered: Hypernym/Hyponym, Meronym/Holonym, Synonym and Co-Hyponym.

For example, G8 (G10 in 6.2.1.1) proposed four nearest words of car as vehicle, automobile, jeep, engine. They are tagged as Hypernym, Synonym, Co-hyponym and Meronym, respectively. Out of 250 words, for each, all the nearest words are detected and manually evaluated. Table 6.6 shows the tendencies of each group to each semantic relation. For example, G1 and G7 are the most productive for meronym/holonym relations with 49.3%. G4 and G7 are most successful dependency relations for synonym extraction. Thus, for each target word in 250, the candidates proposed by only G4 and G7 are taken into consideration. The output of this process gives us up to 12 synonym candidates for 250 target words as a training data.

Table 6.6 Tendencies of groups to semantic relations in percentage

Groups	**Synonym**	**Meronym/Holonym**	**Co-Hyponym**	**Hypernym/Hyponym**
G1	17.7	49.3	22.7	10.1
G2	16.6	48.1	31.4	3.7
G3	16.2	37.8	24.3	21.6
G4	**25**	40.6	21.8	12.5
G5	19.4	35.8	29.8	14.9
G6	21.6	43.3	26.6	8.3
G7	**30.3**	49.3	15.1	5
G8	17.6	45.5	23.5	13.2
Avg	20.7	44.5	24.1	10.5

Features from monolingual dictionaries: Dictionaries are the most popular source for acquiring synonymy relation. In this model, dictionary denitions of target/candidate words are incorporated to compute the similarity of words. All the definitions of target/candidate words are accessed through TDK and Wiki. If target word and its potential synonym mutually appears in their definitions, they are labeled as true, otherwise false as a boolean feature. The main problem is the lack of the definition of a word.

Features from WordNet: This phase is divided into two steps: translation of each pair from Turkish to English and features extraction from WordNet::Similarity modules for each target/candidate word pairs.

- **Bilingual on-line dictionary:** Using bilingual dictionaries is another method for extracting semantic relations, especially synonymy relation. We also utilized bilingual on-line dictionary, Tureng[3]. Target synonym and each candidate synonym words are translated from Turkish to English to exploit the modules in WordNet resources. For example, for a given target word and its all candidates in Turkish are translated into English as follows:

  birey: ilişki insan sermaye performans üniversite yurttaş tempo vatandaş kişi toplum

  individual: affair mortal funds performance university citizen tempo national person collectivity

  Examples of target words and its potentail synonym words in Turkish-English are given in Table B.1.

- **WordNet Corporation**: WordNet::Similarity package [20] is freely available software package which contains a variety of semantic similarity and relatedness measures based on WordNet. It supports the measures of Hirst-St.Onge (HSO), Jiang-Conrath(JCN), Leacock-Chodorow (LCH), Lin(LIN), Banerjee-Pedersen(LESK), Patwardhan-Pedersen(PATH), Resnik(RES), Wu-Palmer(WUP). Some vector modules are also taken into account. All these modules are explained in Section 3.4.1. While some similarity measures are based on path lengths between concepts, some are based on information content. Measures of similarity are based on IS-A hierarchy and show how much two concepts are similar/related. So that the lexical database WordNet is convenient for similarity measures. Target/candidate pairs are given to all modules and their similarities are computed. Some instances failed due to that the modules could not evaluate any results for them. The scores of all the modules are normalized between 0-1 and sum of all normalized scores is also kept as an additional potential feature, namely TOTAL.

Synonym Classification for Model-2: Target class was labeled as SYN/NONSYN by using Monolingual on-line synonym dictionaries. Synonymy of all pairs is mutually checked and tagged. Alongside target class, all features explained above are computed for all pairs/instances. All computed scores of the pairs are kept as a training set. We

[3] http://tureng.com: The Tureng Dictionary is an on-line dictionary service

designed a procedure to produce a list of positive and negative examples of pairs, where all the data are used as training data for machine learning algorithm. While most of attributes contain real value between 0-1, a few of them contain Boolean data. The data is considered to exhibit linear dependency. As a consequence of linearity, logistic regression is again simple approach for such a classification. The most suitable algorithm, logistic regression, can easily be used for binary classification in the domains with numeric attributes and nominal target class.

6.2.2.2 Results and Evaluation of Model-2

Evaluation of attributes can be simply done by looking at their IG scores. Because we accepted that each attribute has a linear relationship with target class. All attributes but co-occurrence provide a positive indicator to detect synonymy. Table 6.7 shows the information gain results. At first glance, WordNet and dictionary-based similarities seem to show good performance. All attributes regarding corpus-based similarities take place towards to the end of list as shown in Table 6.7. Among corpus-based attributes the most important dependency relations are G4 and G7. Among the semantic relations, meronym and holonym relations have better performance than the others. And co-occurrence measure shows a very slight indicator capacity. Error analysis shows that some semantic relations such as hypernymy suffer from the sparse data and the lack of evidence of the words.

The data has imbalance characteristics where the ratio of positive (176 instances) and negative (1098 instances) label is 16% as shown in Table 6.8. This makes the problem harder. Success rate misleadingly shows a good score such as 95.2% as in the Table 6.9. Therefore, we should take F-measure of synonym value of the class into account. Taking the all attributes as features set of training data, the success rate is 95.2%, F-measure for synonymy is 81.3%. When running the model only with WordNet similarity scores, the machine learning algorithms performs F-measure score of 68.2% as shown in Table 6.10. The most successful algorithms of WordNet are LCH and WUP. They are based on depth and shortest-path approaches. Although WordNet has a good performance, it is two times more expensive than other dictionary-based approaches.

Table 6.7 Information gain (IG) of each feature with its type

IG	Features	Types	IG	Features	Types

0.28665	TOTAL	WordNet	0.08787	G7	Corpus
0.28288	LCH	WordNet	0.07007	G4	Corpus
0.28198	WUP	WordNet	0.06929	Meronym	Corpus
0.2751	LIN	WordNet	0.06105	Holonym	Corpus
0.27379	PATH	WordNet	0.0558	G8	Corpus
0.26683	VECTOR	WordNet	0.0419	G1	Corpus
0.26663	HSO	WordNet	0.02336	G2	Corpus
0.25241	TDK-OR-WIKI	Dictionary	0.00816	Co-coccurence	Corpus
0.23395	RES	WordNet	0	G6	Corpus
0.23025	TDK	Dictionary	0	G5	Corpus
0.21854	VECTOR_PAIR	WordNet	0	G3	Corpus
0.21564	LES	WordNet	0	Hypernym	Corpus
0.20769	JCN	WordNet	0	Hyponym	Corpus
0.11425	WIKI	Dictionary			

Table 6.8 Confusion matrix

	NS	**S**
NS	1080	18
S	43	133

Because the words need to be translated into English, afterwards their similarities are measure through WordNet packages and so on. The second weakest point of it is the selection of the first translation of a given word. The other translations and senses are ignored. WordNet needs to be used due to the lack of Turkish WordNet. That lack makes the model more expensive.

Likewise, dictionary-based approach gives following results. The success is 94% and F-measure is 74%. The details can be seen in Table 6.11. It is the most naive approach; the definitions of two words are automatically retrieved from dictionary and their mutual absences are utilized. However, the approach can be considered costly due its nature, unless we dump all content of the dictionaries.

As usual, corpus-based model has a weak performance with F-measure of 41.7%. Table 6.12 covers the all performance measurements for corpus-based model. The main reason of the failure is that some word pairs cannot be represented in corpus-based framework. For example, production capacity of hypernym/hyponymy relation is limited due to that not every pairs are matched by LSP designed for hypernymy. Same reason holds true for other semantic and dependency relation.

In terms of time-cost, the cheapest features are corpus-based ones. The dependency relations are easily extracted from corpus by LSP. Among the semantic relations, meornym/holonym and hyponym/hypernym have a high time complexity. Moreover

hypernymy/hyponymy relation does not return good gain due to the lack of representation of an arbitrary word. Candidate selection phase exploits only dependency relation, since that these relations are easily captured by system. Other corpus-based features are discarded due to several reasons. The most important one is the production capacity of those relations. Hypernymy relation or Meronymy relations do not guarantee returning corresponding result for any given word. We applied a variety of dependency relations. The most effective relations found are G4 and G7. Surprisingly, both relations are the alternations of adjective- modifier patterns. Here, we can conclude that the patterns of adjective modification are very useful to disclose important characteristics of words.

Table 6.9 Precision, recall and F-measure of all attributes

	Precision	**Recall**	**F-Measure**
Class-NS	96.2	98.4	97.3
Class-S	88.1	75.6	81.3
Weighted Average	95.1	95.2	95.1

Table 6.10 Precision, recall and F-measure of features from WordNet

	Precision	**Recall**	**F-Measure**
Class-NS	93.7	97.7	95.7
Class-S	80.6	59.1	68.2
Weighted Average	91.9	92.4	91.9

Table 6.11 Precision, recall and F-measure of features from dictionary definitions

	Precision	**Recall**	**F-Measure**
Class-NS	94.4	98.8	96.6
Class-S	89.6	63.6	74.4
Weighted Average	93.8	94	93.5

Table 6.12 Precision, recall and F-measure of features from corpus

	Precision	**Recall**	**F-Measure**
Class-NS	89.7	97.7	93.5
Class-S	67.9	30.1	41.7
Weighted Average	86.7	88.4	86.4

CHAPTER 7

CONCLUSION

Acquisition of semantic relation is important topic in NLP. There exist a few studies to extract semantic relations for Turkish and most of them based on dictionary definitions. In this study, we presented models for automatic and semi-automatically extracting semantic relation from Turkish corpus. We focused on only hyponym/hypernym, meronym/holonym and synonym relation because of dealing with nouns.

In this study, different models are proposed for each semantic relation. Hence one method can not be available for all relations and does not give expected results. As a result, we concluded each relation with different parts.

For hyponym/hypernym, we presented two different methods for acquiring hyponyms of a given hypernym. The former depends on statistical elimination, latter one relies on statistical expansion. Both model starts with same steps.

First step is relying on LSP and second step is elimination with some assumptions. The model applied the pattern-based method to extract an initial list. We showed that an algorithm based on a particular LSP can retrieve a significant number of hyponymy relations with some assumptions that are particularly applicable to agglutinative language such as Turkish.

For this purpose, the most productive and reliable LSP (NPs gibi CLASS) was found to discover an IS-A hierarchy. We observed that hypernym/hyponm pairs are easily extracted by means of this pattern for Turkish Language. In order to get more precision, we designed some elimination criterias. That gave higher precision but also a limited number of pairs with low recall.

After applying pattern matching and assumptions, the resulting hyponym list can contain many errors or incorrect instances. To eliminate such cases, we applied a model as named Model-1 based semantic similarity using second-order word representation and cosine similarity measurement. The objective is to get better relevance and more precise results.

We observed that semantic similarity measurement in second order word space gives successful results. We used a corpus size of 490 M words in which it is easy to match the patterns. Here, we succeeded in increasing the precision without decreasing the recall. Looking at the Table 4.2 for each case, while recall values slightly decrease or remain the same, precision scores get better. After semantic similarity based improvement, the average precision is increased by 20%. However, the recall values are decreased by 10%. These results indicate that our methodology can robustly capture hyponymy relationships for Turkish.

We concluded that pattern frequency, document frequency and semantic similarity score are the main properties of hyponym candidates. There exist rules that can successfully eliminate unsatisfactory candidates. However, bad design of the word dimension can lead to model failure. Since candidates having low document frequency are a major problem, they must be covered with a different approach. Distinctive and related words must be reserved. Since corpus-based approach has some limitations such as sparseness, we planed to use statistical expansion to increase both precision and recall and so applied Model-2.

Same as Model-1, beginning process of the Model-2 for acquisition of hyponym/hypernym relation relies on the LSP and elimination assumptions. In Model-2, we again proposed a fully automatic model based on syntactic patterns and semantic similarity measurements. Instead of elimination, we designed a bootstrapping algorithm which incrementally enlarged the pair list to get more recall and discover more hyponyms. In our more fine-grained word space, the proposed model successfully expanded the list from the seeds. A variety of semantic similarity measurements are used and evaluated.

In this modular system, we conducted several experiments to analyze the IS-A semantic relation and to find the best setup for the model. When we looked at the experiments in Table 4.4, a LSP based approach gave promising precision. This module successfully

built initial seeds. In order to solve the recall problem, we improved the model capacity to discover new candidates. Both graph-based and simple scoring methodologies are applied and we observed that both approaches had a good capacity to get higher recall figures, such as 71.6% and 72.5%.

A real application could be designed as follows: For the sake of simplicity, a simple scoring method with binary weighting would be the best setup with respect to the results. Moreover, the all reliable candidates proposed by the pattern-based method might be used as initial seeds to make the model more robust. The pattern module can be refined to obtain more secure candidates. The number of hyponyms to be extracted can be automatically decided. The output size of the pattern module is a good indicator for a decision.

The results showed that the fully automated model presented in this study successfully discovers IS-A relations by mining a large corpus, without other input. We also implemented and provided a utility program to verify and reproduce the results that we found.

For disclosing meronym/holonym relation, we again utilized and adopted LSPs to Turkish corpus. Two different approaches are considered to prepare patterns; one is based on pre-defined patterns that are taken from literature. Second approach automatically produces patterns by means of bootstrapping method. Prepared patterns fall into two clusters; General patterns and Dictionary-based patterns. In addition to these three clusters, we also used statistical selection on global data obtaining from all results of entire patterns.

After morphologically parsing a huge corpus, all patterns are formed and adopted into the specific regular expressions in accordance with parsed corpus. Each pattern is designed so that we can separately pick up whole and its potential part candidates to be proposed. With a variety of experiments, we addressed some problems, conclude a list of facts and achieve successful results for Turkish meronymy problem. As analyzing general patterns and dictionary-based patterns, it is said that an appropriate pattern design is capable of solving the problem of meronymy.

One of the important challenges addressed is sense ambiguity that increases the error rate. System could suffer from either word sense ambiguity or pattern ambiguity. Second challenge is facing some diffculties due to free word order and morphological

characteristics of Turkish language. Determining window size of a pattern is crucial. The raw text is needed to parse with an accurate morphological parser. Third, error analysis shows that the patterns can also encode other kind of semantic relations rather than meronymy. Another challenge is use of language, because a corpus can includes many types of expression from written to spoken. Thus, even the strongest patterns cannot match due to this factor.

Several significant findings of the study are reported in Section 5.6.7. Some of them can be listed as follows: Even though dictionary-based patterns are so suitable for dictionary-like corpus by definition, they have good and comparable potential to extract part-whole pairs from a corpus. General patterns are slightly better than them. Some particular patterns from both clusters, GP and TDK-P, have a good indicative capacity in terms of production and precision. The approach utilizing bootstrapping first retrieves reliable patterns, then, extracts and proposes some part candidates for a given whole. The successful results of that approach are comparable to other predefined patterns. It has also domain-independent characteristic and good production capacity as well. Therefore it is said that it can be easily applied for other relation problems.

Instead of applying each pattern one by one, all results from entire pattern list are merged as input for statistical methods. The big data, namely global whole-by-part matrix, are measured by means of several statistics such as IG, T-score, etc. The results indicate that it has very similar behavior with bootstrapped pattern, where the results are comparable to predefined list. Moreover statistical selection and bootstrapping have large scale and good production capacity. Production capacity denoted by #ofCpW>1 of a pattern refers to how many cases matched per whole on average. It and success ratio can be combined to evaluate proposed patterns. Even though some patterns seem to have good accuracy, they have less production capacities. Thus, the output of such patterns has limited number of wholes. We evaluated the success of patterns over not only precision but also combined ranking factor taking #ofCpW>1 and success rate as parameters.

No matter to which cluster a pattern belongs, if a pattern can produce higher number of unique wholes, it can show better performance. We checked which pattern characteristic affects success rate (precision) by means of correlation formula. The correlation table at Table 5.13 indicates success of a pattern highly depends on number of producing unique

wholes. Second important attribute of a pattern is average number of cases per whole as indicated in Table 5.13.

Incorrectly selected parts are mostly based on other semantic relations. Error analysis shows us that 13% error rate comes from other semantic relations such as hyponymy, synonymy, on average for predefined list. Similar error rate of 10% originating from other relations is measured in statistical selection. This indicates that patterns can successfully disclose semantic relatedness in a large scope.

As final remark, all experiments indicate that proposed methods have good indicative capacity for solution of the problem addressed, because each method can outperform its corresponding baseline algorithm.

For synonym, we developed two models. In Model-1, synonym pairs are determined on the basis of co-occurrence statistics, semantic and dependency relations within distributional aspect. Contrary to hypernymy and meronymy relation, simply applying LSPs does not extract synonym pairs from a big corpus. Instead, we extracted other semantic relations to ease detection of synonymy. Our methodology relies on some assumptions. One is that the synonym pairs mostly show similar semantic characteristics by definition. They share the same meronym/holonym and hypernym/hyponym relations. Particular LSPs can be used to initiate the acquisition process of those semantic features.

Secondly, a pair of synonym words mostly shares a particular list of governing verbs and modifying adjectives. The more a pair of words are governed by similar verb profile and modified by similar adjectives, the more likely they are synonym. We built 10 groups of syntactic patterns according to their syntactic structures.

To apply machine learning algorithm, three annotators manually and randomly selected 200 synonym pairs and 200 non-synonyms. Non-synonym pairs were especially selected from associated (relevant) pairs such as tree-leaf, apple-orange, school-student. Otherwise, such negative example selection could lead to false inference. The main challenge faced in the experiments is the lack of features of some words due to their corpus evidence. Thus, such instances were eliminated. Remaining instances was classified by the most suitable algorithm which is the logistic regression. It can easily be used for binary classification in domains with numeric attributes and nominal target class.

As long as individual performance of any feature is higher than F-measure of 50%, it is considered as useful features or considered independent feature from target class. The aim was to find out which features are the most informative for detecting synonymy and contribute most to the overall success of the model. When comparing the results, it was clearly observed that the semantic features are notably better than syntactic dependency models in finding true synonyms. The most effective attributes are meronymy and holonymy features with weighted average F-measure of 68.7% and 73.7% respectively. The analysis indicated that the possible reason for the success is sufficiency in the number of cases from which semantic and dependency features are constructed. As a consequence of insuffcient data volume, hypernymy/hyponymy relation is relatively worse than meronymy. Among dependency relations, G1, G4 and G7 outperformed the others. Likewise, it was also observed that sufficiency in the number of cases was the strong factor. After aggregating all useful features, the same learning process was carried out. The aggregated model shows promising results and performance. Regression model achieved an acceptable F-measure of 80.3%.

In Model-2, extracting synonym from corpus is performed by incorporating other resources such as dictionaries, WordNet besides using distributional features of words. In this model, a variety of features from corpus-based to dictionary-based, are incorporated to solve the synonymy extraction problem for Turkish Language. The model is well proposed in a way that for a given word, it automatically produces some candidate words and decides if there is synonymy or not. To build a candidate word list, dependency relations are exploited. The closest K words for a given target word are taken as candidate list. The similarities between target and each candidate are computed by 27 different features with a large variety of measurements. The attributes are evaluated according to their IG scores. The results show that the best features are dictionary-based measurements. All measures from WordNet almost show good performances. Second, the monolingual dictionary definitions are the other significant factors. Third, corpus-based similarities, both dependency relations and semantic relations, have slight impact on success. However, dependency relations are the easiest and inexpensive way to capture candidate words. It has been shown that the patterns of adjective modification are very useful to disclose characteristics of words.

Taking all attributes into account the model gives a performance of F-measure of 81.4%. Whereas the model based on only WordNet similarities perform F-measure of

68%, dictionary-based similarity gives F-measure of 74%. The corpus-based features underperform with F-measure of 41.7%. The main reason of the failure is the production capacity of the some patterns. Thus, not all words are represented with sufficient number of examples.

As a result, main contribution of the study is to become first major corpus-driven attempt to extract semantic relations such as hyponym/hypernym, meronym/holonym and synonym, from Turkish language. Second contribution is to use integrated approaches such as pattern-based method with statistical elimination and expansion, bootstrapping patterns, etc. Third contribution is to use multiple resources and to integrate them such as WordNet, mono/bilingual on-line dictionaries, etc.

Our principal goal is to automatically build a Turkish semantic lexicon including many noun-noun relations such as hypernymy, meronymy, synonymy from only a huge corpus. We have developed some models for hyponym/hypernymy, meronymy/holonym and synonym. We have concluded with a discussion of results and showed the models presented here gives promising results for Turkish text.

REFERENCES

[1] Lyons, J., (1968). Introduction to Theoretical Linguistics, Cambridge University Press, New York.

[2] Lyons, J., (1977). Semantics, Cambridge University Press, New York.

[3] Cruse, D.A., (1986). Lexical Semantics, Cambridge University Press, New York.

[4] Chaffin, R. and Douglas, J.H., (1984). "The Similarity and Diversity of Semantic Relations", Memory and Cognition, 12(2):134-141.

[5] Hearst, M.A., (1992). "Automatic Acquisition of Hyponyms from Large Text Corpora", Proceedings of the 14th International Conference on Computational Linguistics, COLING 1992, 23-28 August 1992, Nantes, France, 539-545.

[6] Riloff, E. and Shepherd, J., (1997). "A Corpus-Based Approach for Building Semantic Lexicons", Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing, 1-2 August 1997, Brown University, Providence, Rhode Island, 109-116.

[7] Lin, D., (1998). "Automatic Retrieval and Clustering of Similar Words", Proceedings of the 17th International Conference on Computational Linguistics, COLING-ACL 1998, 10-14 August 1998, Université de Montréal, Montréal, Quebec, Canada, 768-774.

[8] Girju, R., Beamer, B., Rozovskaya, R. and Bhat, S., (2010). "UIUC: A Knowledge-rich Approach to Identifying Semantic Relations between Nominals", Information Processing Management, 46(5):589-610.

[9] Palmer, F.P., (1981). Semantics, Cambridge University Press, New York.

[10] Murphy, M.L., (2003). Semantic Relation and the Lexicon: Antonymy, Synonymy and other Paradigms, Cambridge University Press, New York.

[11] Landis, T.Y., Herrmann, D.J. and Chan, R., (1987). "Developmental Differences in the Comprehension of Semantic Relations", Zeitschrift für Psychologie mit Zeitschrift für angewandte Psychologie, 195(2):129-139.

[12] Lauer, M., (1995). Designing Statistical Language Learners: Experiments on Noun Compounds, Ph.D. Thesis, Macquarie University, Sydney, Australia.

[13] Nastase, V. and Szpakowicz, S., (2003). "Exploring Noun-Modifier Semantic Relations", Proceedings of the 5th International Workshop on Computational Semantics, 15-17 January 2003, Tilburg, Netherlands, 285-301.

[14] Moldovan, D., Badulescu, A., Tatu, M., Antohe, D. and Girju, R., (2004). "Models for the Semantic Classification of Noun Phrases", The Human

Language Technology (HLT)/North American Chapter of the Association for Computational Linguistics, 2-7 May 2004, Boston, Massachusetts, 60-67.

[15] Girju, R., Moldovan, D., Tatu, M. and Antohe, D., (2005). "On the Semantics of Noun Compounds", Computer Speech and Language, 19(4):479-496.

[16] Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D. and Miller, K.J., (1993). "Introduction to WordNet: An On-line Lexical Database", International Journal of Lexicography, 3(4):235-244.

[17] Finin, T.W., (1980). "The Semantic Interpretation of Nominal Compounds", Association for the Advancement of Artificial Intelligence, 18-21 August 1980, Stanford, California, 310-312.

[18] Vanderwende, L., (1994). "Algorithm for Automatic Interpretation of Noun Sequences", Proceedings of the 15th International Conference on Computational Linguistics, COLING 1994, 5-9 August 1994, Kyoto, Japan, 782-788.

[19] Lauer, M., (1995). "Corpus Statistics Meet the Noun Compound: Some Empirical Results", 33rd Annual Meeting of the Association for Computational Linguistics, 26-30 June 1995, MIT, Cambridge, Massachusetts, USA, 47-54.

[20] Rosario, B. and Hearst, M., (2001). "Classifying the Semantic Relations in Noun Compounds", Conference on Empirical Methods in Natural Language Processing, 3-4 June 2001, Pittsburgh, Pennsylvania, 82-90.

[21] Rosario, B., Hearst, M.A. and Fillmore, C., (2002). "The Descent of Hierarchy, and Selection in Relational Semantics", Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 6-12 July 2002, Philadelphia, PA, USA, 247-254.

[22] Lapata, M., (2000). "The Automatic Interpretation of Nominalizations", American Association for Artificial Intelligence, AAAI-00, 1-3 August 2000, Austin, Texas, US, 716-721.

[23] Lapata, M., (2002). "The Disambiguation of Nominalizations", Computational Linguistics, 28(3):357-388.

[24] Turney, P. D. and Litman, M., (2005). "Corpus-Based Learning of Analogies and Semantic Relations", Machine Learning, 60:(1-3).

[25] Turney, P., (2006). "Expressing Implicit Semantic Relations without Supervision", 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, 17-21 July 2006, Sydney, Australia, 313-320.

[26] Girju, R., Giuglea, A., Olteanu, M., Fortu, O., Bolohan, O. and Moldovan, D., (2004). "Support Vector Machines Applied to the Classification of Semantic Relations Innominalized Noun Phrases", Proceedings of the HLT/NAACL Workshop on Computational Lexical Semantics, 2-7 May 2004, Boston, MA, US, 68-75.

[27] Girju, R., Nakov, P., Nastase, V., Szpakowicz, S., Turney, P. and Yuret, D., (2007). "SemEval-2007 Task 04: Classification of Semantic Relations between Nominals", The 4th International Workshop on Semantic Evaluations (SemEval), Association for Computational Linguistics, 23-24 June 2007, Prague, Czech Republic, 13-18.

[28] Panchenko, A., (2012). Similarity Measures for Semantic Relation Extraction. Ph.D. Thesis, Université Catholique de Louvain & Bauman Moscow State Technical University, Moscow, Russia.

[29] Miller, G.A., (1990). "WordNet: An On-line Lexical Database", International Journal of Lexicography, 3(4):235-312.

[30] Miller, G.A., (1995). "WordNet: A Lexical Database for English", Communications of the ACM, 38(11):39-4.

[31] Fellbaum, C., (1998). WordNet: An Electronic Lexical Database, MIT Press, Cambridge.

[32] Igo, S.P. and Riloff, E., (2009). "Corpus-based Semantic Lexicon Induction with Web-based Corroboration", Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, 31 May - 5 June 2009, Boulder, Colorado, USA, 18-26.

[33] Alshawi, H., (1987). "Processing Dictionary Definitions with Phrasal Pattern Hierarchies", American Journal of Computational Linguistics, 13(3-4):195-202.

[34] Markowitz, J., Ahlswede, T. and Evens, M., (1986). "Semantically Significant Patterns in Dictionary Definitions", Proceedings of 24th Annual Meeting of the Association for Computational Linguistics, 10-13 July 1986, Columbia University, New York, USA, 112-119.

[35] Jensen, K., and Binot, J., (1987). "Disambiguating Prepositional Phrase Attachments by Using On-Line Dictionary Definitions", American Journal of Computational Linguistics, 13(3-4):251-260.

[36] Pantel, P. and Pennacchiotti, M., (2006). "Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations", Proceeding of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, 17-21 July 2006, Sydney, Australia, 113-120.

[37] Giovannetti, E., Marchi, S., and Montemagni, S., (2008). "Combining Statistical Techniques and Lexico-Syntactic Patterns for Semantic Relations Extraction from Text", Proceedings of the 5th Workshop on Semantic Web Applications and Perspectives, SWAP 2008, 15-17 December 2008, Rome, Italy.

[38] Ruiz-Casado, M., Alfonseca, E. and Castells, P., (2007). "Automatising the Learning of Lexical Patterns: An application to the Enrichment of WordNet by Extracting Semantic Relationships from Wikipedia", Data Knowledge Engineering, 61(3):484-499.

[39] Turney, P.D., (2008). "A Uniform Approach to Analogies, Synonyms, Antonyms, and Associations", Proceedings of the 22nd International Conference on Computational Linguistics, 18-22 August 2008, Manchester, UK, 905-912.

[40] Roark, B. and Charniak, E., (1998). "Noun-phrase Co-occurrence Statistics for Semi-automatic Semantic Lexicon Construction", 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference

on Computational Linguistics, COLING-ACL 1998, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada, 1110-1116.

[41] Widdows, D. and Dorow, B., (2002). "A Graph Model for Unsupervised Lexical Acquisition", 19th International Conference on Computational Linguistics, COLING 2002, 24 August – 1 September 2002, Taipei, Taiwan, 1093-1099.

[42] Phillips, W. and Riloff, E., (2002). "Exploiting Strong Syntactic Heuristics and Co-training to Learn Semantic Lexicons", Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, 6-7 July 2002, Philadephia, Pennsylvania, USA, 125-132.

[43] Riloff, E. and Jones, R., (1999). "Learning Dictionaries for Information Extraction by Multi-level Bootstrapping", The 16th National Conference on Artificial Intelligence, 18-22 July 1999, Orlando, Florida, US, 1044-1049.

[44] Brill, E. and Resnik, P., (1994). "A Rule-based Approach to PP Attachment Disambiguation", 15th International Conference on Computational Linguistics, COLING 1994, 5-9 August 1994, Kyoto, Japan, 998-1004.

[45] Lapata, M. and Keller, F., (2004). "The Web as a Baseline: Evaluating the Performance of Unsupervised Web-based Models for a Range of NLP tasks", Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, 2-7 May 2004, Boston, MA, USA, 121-128.

[46] Davidov, D. and Rappoport, A., (2008). "Classification of Semantic Relationships between Nominals Using Pattern Clusters", Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, 15-20 June 2008, Columbus, Ohio, USA, 227-235.

[47] Nakov, P. and Hearst, M., (2006). "Using Verbs to Characterise Noun-Noun Relations", Artificial Intelligence: Methodology, Systems, and Applications. 12th International Conference, AIMSA 2006, 12-15 September 2006, Varna, Bulgaria, 233-244; Editors: Euzenat, J. and Domingue, J. (2006). LNCS 4183, Berlin, Germany, Springer.

[48] Panchenko, A., (2011). "Comparison of the Baseline Knowledge-, Corpus-, and Web-based Similarity Measures for Semantic Relations Extraction", Proceedings of the GEMS 2011 Workshop on Geometrical Models of Natural Language Semantics, EMNLP 2011, 31 July 2011, Edinburgh, Scotland, UK, 11-21.

[49] Bilgin, O., Cetinoğlu, Ö. and Oflazer, K., (2004). "Building a WordNet for Turkish", Journal of Information Science and Technology, 7(1-2):163-172.

[50] Amasyalı, M.F., (2005). "Türkçe Wordnet'in Otomatik Olarak Oluşturulması", IEEE 13. Sinyal İşleme ve İletişim Uygulamaları Kurultayı, SIU-2005, 16-18 May 2005, Kayseri, Turkey.

[51] Yazıcı, E. and Amasyalı, M.F., (2011). Automatic Extraction of Semantic Relationships Using Turkish Dictionary Definitions, EMO Bilimsel Dergi, İstanbul.

[52] Güngör, O. and Güngör, T., (2007). "Türkçe Bir Sözlükteki Tanımlardan Kavramlar Arasındaki Üst-kavram İlişkilerinin Çıkarılması", Akademik Bilişim Konferansı.

[53] Serbetçi, A., Orhan, Z. and Pehlivan, I., (2011). "Extraction of Semantic Word Relations in Turkish from Dictionary Definitions", Proceedings of the ACL 2011 Workshop on Relational Models of Semantics, RELMS 2011, Association for Computational Linguistics, 23 June 2011, Portland, Oregon, USA, 11-18.

[54] Orhan, Z., Pehlivan, I., Uslan, V. and Onder, P., (2011). "Automated Extraction of Semantic Word Relations in Turkish Lexicon", Mathematical and Computational Applications, 16(1):13-22.

[55] Sak, H., Güngör, T. and Saraçlar, M., (2008). "Turkish Language Resources: Morphological Parser, Morphological Disambiguator and Web Corpus", The 6th International Conference on Natural Language Processing, GOTAL 2008, 27-27 August 2008, Gothenburg, Sweden, 417-427; Editors: Nordström, B. and Ranta, A. LNCS 5221, Berlin, Germany, Springer.

[56] Zellig, H., (1954). "Distributional Structure", Word, 10(23):146-162.

[57] Pedersen, T., Patwardhan, S. and Michelizzi, J., (2004). "WordNet:Similarity: Measuring the Relatedness of Concepts". Proceedings of the 19th National Conference on Artificial Intelligence, 16th Conference on Innovative Applications of Artificial Intelligence, 25-29 July 2004, San Jose, California, USA, 38-41.

[58] Resnik, P., (1995). "Using Information Content to Evaluate Semantic Similarity", Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI 95, 20-25 August 1995, Montréal Québec, Canada, 448-453.

[59] Lin, D., (1998). "An Information-Theoretic Definition of Similarity", Proceedings of the 15th International Conference on Machine Learning, ICML 1998, 24-27 July 1998, Madison, Wisconsin, USA, 296-304.

[60] Jiang, J. and Conrath, D., (1997). "Semantic Similarity based on Corpus Statistics and Lexical Taxonomy", Proceedings of International Conference on Research in Computational Linguistics, 22-24 August 1997, Taipei, Taiwan, 19-33.

[61] Leacock, C. and Chodorow, M., (1998). "Combining Local Context and WordNet Similarity for Word Sense Identification", WordNet: An Electronic Lexical Database, 265-283; Editor: Fellbaum, C. (1998), MIT Press, Cambridge.

[62] Wu, Z. and Palmer, M., (1994). "Verb Semantics and Lexical Selection", 32nd Annual Meeting of the Association for Computational Linguistics, 27-30 June 1994, Las Cruces, New Mexico, USA, 133-138.

[63] Hirst, G. and St-Onge, D., (1998). "Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms", WordNet: An Electronic Lexical Database, 305-332; Editor: Fellbaum, C. (1998). MIT Press, Cambridge.

[64] Banerjee, S. and Pedersen, T., (2002). "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet", Computational Linguistics and Intelligent Text Processing, 3rd International Conference, CICLing 2002, 17-23 February 2002, Mexico City, Mexico, 136-145.

[65] Schütze, H., (1998). "Automatic Word Sense Discrimination", Computational Linguistics, 24(1):97-123.

[66] Patwardhan S. and Pedersen T., (2006). "Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts", 11th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, 3-7 April 2006, Trento, Italy, 1-8.

[67] Patwardhan, S., (2003). Incorporating Dictionary and Corpus Information into a Vector Measure of Semantic Relatedness, Master Thesis, University of Minnesota, Minneapolis, MN, USA.

[68] Manning, C.D., Raghavan, P. and Scühtze, H., (2008). Introduction to Information Retrieval, Cambridge University Press, New York.

[69] Pedersen, T., Banerjee, S., Kohli, S., Joshi, M., McInnes, B.T. and Liu, Y., (2011). "The Ngram Statistics Package (Text:NSP) - A Flexible Tool for Identifying Ngrams, Collocations, and Word Associations", Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World, MWE 2011, Association for Computational Linguistics, 23 June 2011, Portland, Oregon, US, 131-133.

[70] Salton, G., Wong, A. and Yang, C.S., (1975). "A Vector Space Model for Automatic Indexing", Communications of the ACM, 18(11):613-620.

[71] Cruse, D., (2002). "Hyponymy and Its Varieties", Information Science and Knowledge Management, 35-50; Editors: Green, R., Bean, C.A. and Myaeng, S.H. (2002).The Semantics of Relationships, Springer Netherlands.

[72] Tversky, B., (1990). "Where Partonomies and Taxonomies Meet", 291-306; Editors: Tsohatzidis, S.L. Meanings and prototypes: Studies in Linguistic Categorizaition, Routledge, New York.

[73] Cruse, D.A., (2004). Meaning in Language. An Introduction to Semantics and Pragmatics, Oxford University Press, Oxford.

[74] Alfonseca, E. and Manandhar, S., (2002). "Improving an Ontology Refinement Method with Hyponymy Patterns", Proceedings of the 3rd International Conference on Language Resources and Evaluation, LREC 2002, 29-31 May 2002, Las Palmas, Canary Islands, Spain.

[75] Mann, G.S., (2002). "Fine-Grained Proper Noun Ontologies for Question Answering", 19th International Conference on Computational Linguistics, COLING 2002, 24 August – 1 September 2002, Howard International House and Academia Sinica, Taipei, Taiwan, 1-7.

[76] Ruiz-Casado, M., Alfonseca, E. and Castells, P., (2005). "Automatic Extraction of Semantic Relationships for WordNet by Means of Pattern Learning from Wikipedia", 10th International Conference on Applications of Natural Language to Information Systems, NLDB 2005, 15-17 June 2005, Alicante, Spain, 67-79; Editors: Montoyo, A., Muñoz, R. and Métais, E. (2005), LNCS 3513, Springer, Berlin, Heidelberg.

[77] Snow, R., Jurafsky, D. and Ng, A., (2005). "Learning Syntactic Patterns for Automatic Hypernym Discovery", Advances in Neural Information Processing Systems 18, Neural Information Processing Systems, NIPS 2005, 5-9 December 2005, Vancouver, British Columbia, Canada, 1297-1304.

[78] Ando, M., Sekine, S.and Ishizaki, S., (2004). "Automatic Extraction of Hyponyms from Japanese Newspapers Using Lexico-syntactic Patterns", In IPSJ SIG Technical Report, (98):77-82.

[79] Sombatsrisomboon, R., Matsuo, Y. and Ishizuka, M., (2003). "Acquisition of Hypernyms and Hyponyms from the WWW", Proceedings of the 2nd International Workshop on Active Mining 2003, 28 October 2003, Maebashi, Japan.

[80] Rydin, S., (2002). "Building a Hyponymy Lexicon with Hierarchical Structure", Proceedings of the ACL-02 workshop on Unsupervised Lexical Acquisition, SIGLEX 2002, 7-12 June 2002, Philadelphia, 26-33.

[81] Elghamry, K., (2008). "Using the Web in Building a Corpus-Based Hypernymy-Hyponymy Lexicon with Hierarchical Structure for Arabic", The 6th International Conference on Informatics and Systems, INFOS2008, 27-29 March 2008, Cairo, Egypt.

[82] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S. and Yates, A., (2005). "Unsupervised Named-Entity Extraction from the Web: An Experimental Study", Artificial Intelligent, 165(1):91-134.

[83] Pasca, M., (2004). "Acquisition of Categorized Named Entities for Web Search", Proceedings of the 2004 ACM CIKM International Conference on Information and Knowledge Management, 8-13 November 2004, Washington, DC, USA, 137-145.

[84] Sang, E.F.T.K. and Hofmann, K., (2007). "Automatic Extraction of Dutch Hypernym-Hyponym Pairs", Proceedings of the 17th Meeting of Computational Linguistics, 12 January 2007, Netherlands.

[85] Sang, E.F.T.K., (2007). "Extracting Hypernym Pairs from the Web", Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, 23-30 June 2007, Prague, Czech Republic, 165-168.

[86] Ritter, A., Soderland, S. and Etzioni, O., (2009). "What Is This, Anyway: Automatic Hypernym Discovery", The Association for the Advancement of Artificial Intelligence, AAAI Spring Symposium, 23-25 March 2009, Palo Alto, California, USA, 88-93.

[87] Shinzato, K. and Torisawa, K., (2004). "Acquiring Hyponymy Relations from Web Documents", Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, 2-7 May 2004, Boston, Massachusetts, USA, 73-80.

[88] Caraballo, S.A., (1999). "Automatic Construction of a Hypernym-Labeled Noun Hierarchy from Text", 27th Annual Meeting of the Association for Computational Linguistics, 20-26 June 1999, Univeristy of Maryland, College Park, Maryland, USA, 120-126.

[89] Pantel, P. and Ravichandran, D., (2004). "Automatically Labeling Semantic Classes", Human Language Technology Conference of the North American

Chapter of the Association for Computational Linguistics, 2-7 May 2004, Boston, Massachusetts, USA, 321-328.

[90] Cederberg, S. and Widdows, D., (2003). "Using LSA and Noun Coordination Information to Improve the Precision and Recall of Automatic Hyponymy Extraction", Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003, 27 May - 1 June 2003, Edmonton, Canada, 111-118.

[91] Kozareva, Z., Riloff, E. and Hovy, E.H., (2008). "Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs", Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, 15-20 June 2008, Columbus, Ohio, USA, 1048-1056.

[92] Hovy, E.H., Kozareva, Z. and Riloff, E., (2009). "Toward Completeness in Concept Extraction and Classification", Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, 948-957.

[93] Kozareva, Z. and Hovy, E., (2010). "A Semi-supervised Method to Learn and Construct Taxonomies Using the Web", Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, 1110-1118.

[94] Sumida, A. and Torisawa, K., (2008). "Hacking Wikipedia for Hyponymy Relation Acquisition", Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP), 7-12 January 2008, Hyderabad, India, 883-888.

[95] Sumida, A., Yoshinaga, N. and Torisawa, K., (2008). "Boosting Precision and Recall of Hyponymy Relation Acquisition from Hierarchical Layouts in Wikipedia", Proceedings of the 6th International Language Resources and Evaluation (LREC), 28-30 May 2008, Marrakech, Morocco.

[96] Herbelot, A. and Copestake, A., (2006). "Acquiring Ontological Relationships from Wikipedia Using RMRS", Proceeding of the ISWC 2006 Workshop on Web Content Mining with Human Language Technologies, 5-9 November 2006, Athens, GA, USA.

[97] Cruse, A.D., (2003). The Handbook of Linguistics, The Lexicon, Blackwell Handbooks in Linguistics, Wiley.

[98] Keet, C.M. and Artale, A., (2008). "Representing and Reasoning Over a Taxonomy of Part-Whole Relations", Applied Ontology 3(1-2):91-110.

[99] Pribbenow, S., (2002). "Meronymic Relationships: From Classical Mereology to Complex Part-whole Relations", 35-50; Editors: Green, R., Bean, C.A. and Myaeng, S.H., The Semantics of Relationships, Springer, Netherlands.

[100] Croft, W. and Cruse, D., (2004). Cognitive Linguistics, Cambridge Textbooks in Linguistics. Cambridge University Press, New York.

[101] Maedche, A., (2001). "Ontology Learning for the Semantic Web", IEEE Intelligent Systems, 16(2):72-79.

[102] Casati, R. and Varzi, A., (1999). Parts and Places: The Structures of Spatial Representation, A Bradford book, MIT Press, Cambridge.

[103] Simons, P., (1987). Parts: A Study in Ontology, Clarendon Press, Isle of Wight.

[104] Gerstl, P. and Pribbenow, S., (1995). "Midwinters, End Games, and Body Parts: A Classification of Part-whole Relations", International Journal of Human Computer Studies, 43(5-6):865–889.

[105] Iris, M.A., Litowitz, B.E. and Evens, M., (1989), "Problems of The Part-whole Relation", 261-288; Editor: Evens, M. In Relational Models of the Lexicon, Cambridge University Press, New York.

[106] Winston, M.E., Chan, R. and Herrmann, D., (1987). "A Taxonomy of Part-Whole Relations", Cognitive Science, 11(4):417-444.

[107] Markowitz, J.A., Nutter, J.T. and Evens, M.W., (1992). "Beyond Is-a and Part-whole: More Semantic Network Links", Computers and Mathematics with Applications, 23(6-9):377-390.

[108] Odell, J.J., (1994). "Six Different Kinds of Composition", Journal of Object-Oriented Programming, 5(8):10-15.

[109] Wanner, L., (1996). Lexical Functions in Lexicography and Natural Language Processing, John Benjamins, Amsterdam.

[110] Artale, A., Franconi, E., Guarino, N. and Pazzi, L., (1996). "Part-whole Relations in Object-centered Systems: An Overview", Data and Knowledge Engineering, 20(3):347-383.

[111] Keet, C.M., (2006). "Part-whole Relations in Object-role Models", 1118-1127; Editors: Meersman, R., Tari, Z., Herrero, P. OTM Workshops (2), LNCS 4278, Springer Berlin Heidelberg.

[112] Kuczora, P.W. and Cosby, S.J., (1989). "Implementation of Meronymic (Part-whole) Inheritance for Semantic Networks", Knowledge-Based Systems 2(4):219-227.

[113] Girju, R., Badulescu, A. and Moldovan, D., (2003). "Learning Semantic Constraints for the Automatic Discovery of Part-whole Relations", Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, May 27- June 1 2003, Edmonton, Canada 1-8.

[114] Ittoo, A. and Bouma, G., (2010). "On Learning Subtypes of The Part-whole Relation: Do Not Mix Your Seeds", Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL'10, Association for Computational Linguistics, 11-16 July 2010, Uppsala, Sweden, 1328-1336.

[115] Van, H.W.R., Kolb, H. and Schreiber, G., (2006). "A Method for Learning Part-whole Relations", 723-735; Editors: Cruz, I.F., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., and Aroyo, L. International Semantic Web Conference, LNCS 4273, Springer.

[116] Buscaldi, D, Rosso, P. and Arnal, E.S., (2005). "A WordNet-based Query Expansion Method for Geographical Information Retrieval". In Working Notes for the CLEF Workshop, Vienna, Austria.

[117] Schulz, S. and Hahn, U., (2005). "Part-whole Representation and Reasoning in Formal Biomedical Ontologies", Artificial Intelligence in Medicine, 34(3):179-200.

[118] Berland, M. and Charniak, E., (1999). "Finding Parts in Very Large Corpora", Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, 20-26 June 1999, College Park, MD, USA, 57-64.

[119] Girju, R., Badulescu, A. and Moldovan, D., (2006). "Automatic Discovery of Part-whole Relations", Computational Linguistics, 32(1):83-135.

[120] Ittoo, A., Bouma, G., Maruster, L. and Wortmann, H., (2010). "Extracting Meronymy Relationships from Domain Specific, Textual Corporate Databases", 48-59; Editors: Hopfe, C.J., Rezgui, Y., M´etais, E., Preece, A.D. and Li, H. NLDB, LNCS 6177, Springer.

[121] Xinyu, C., Cungen, C., Shi, W. and Han, L., (2008). "Extracting Part-whole Relations from Unstructured Chinese Corpus", Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery, 18-20 October 2008, Shandong, China, 175-179.

[122] Quinlan J.R., (1993). C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco.

[123] Beckwith, R. and Miller, G.A., (1990). "Implementing a Lexical Network", International Journal of Lexicography, 4(3):302-312.

[124] Mandala, R., Tokunaga, T. and Tanaka, H., (1999), "Combining Multiple Evidence from Different Types of Thesaurus for Query Expansion", Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 15-19 August 1999, Berkeley, CA, USA, 191-197.

[125] Radev, D., Qi, H., Zheng, Z., Zhang, Z., Fan, W., Goldensohn, S. and Prager, J., (2001). "Mining the Web for Answers to Natural Language Questions", Proceedings of the 10th International Conference on Information and Knowledge Management, 06-11 November 2000, McLean, VA, USA, 143-150.

[126] Kiyota, Y., Kurohashi, S. and Kido, F. (2002). "Dialog Navigator: A Question Answering System Based on Large Text Knowledge Base", Proceedings of the 19th International Conference on Computational Linguistics, 24 August- 1 September, Taipei, Taiwan.

[127] Bai, J., Song, D., Bruza, P., Nie, J.Y. and Cao, G., (2005). "Query Expansion Using Term Relationships in Language Models for Information Retrieval", Proceedings of the 14th ACM International Conference on Information and Knowledge Management, 31 October- 5 November 2005, Bremen, Germany.

[128] Riezler, S., Liu, Y. and Vasserman, A., (2008). "Translating Queries into Snippets for Improved Query Expansion", Proceedings of the 22nd International Conference on Computational Linguistics, 18-22 August 2008, Manchester, UK, 737-744.

[129] Inkpen, D., (2007). "A Statistical Model for Near-synonym Choice", ACM Transactions on Speech and Language Processing, 4(1):1-17.

[130] Barzilay, R. and Elhadad, M., (1997). "Using Lexical Chains for Text Summarization", Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization, 1 July 1997, Madrid, Spain, 10-17.

[131] Inkpen, D.Z. and Hirst, G., (2003). "Near-synonym Choice in Natural Language Generation", 141-152; Editors: Nicolov, N., Bontcheva, K., Angelova, G. Mitkov, R. RANLP, John Benjamins, Amsterdam/Philadelphia.

[132] McCarthy, D. And Navigli, R., (2009). "The English Lexical Substitution Task", Language Resources and Evaluation, 43(2):139-159.

[133] Mirkin, S., Dagan, I. and Geffet, M., (2006). "Integrating Pattern-Based and Distributional Similarity Methods for Lexical Entailment Acquisition", Proceedings of the COLING/ACL on Main Conference Poster Sessions, ACL, 17-21 July 2006, Sydney, Australia, 579-586.

[134] Lin, D., Zhao, S., Qin, L. and Zhou, M., (2003). "Identifying Synonyms Among Distributionally Similar Words", Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI-2003, 8-15 August 2003, Acapulco, Mexico, 1492-1493.

[135] Wang, W., Thomas, C., Sheth, A.P. and Chan, V., (2010). "Pattern-based Synonym and Antonym Extraction", Proceedings of the 48th Annual Southeast Regional Conference, 15-17 April 2010, Oxford, MS, USA.

[136] Hagiwara, M.A., (2008). "Supervised Learning Approach to Automatic Synonym Identification Based on Distributional Features", Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Student Research Workshop, ACL, 15-20 June 2008, Colombus, OH, USA, 1-6.

[137] Heylen, K., Peirsman, Y., Geeraerts, D. and Speelman, D., (2008). "Modelling Word Similarity: an Evaluation of Automatic Synonymy Extraction Algorithms", Proceedings of 6th International Conference on Language Resources and Evaluation, LREC, 28-30 May 1998, Marrakech, Morocco.

[138] Hindle, D., (1990). "Noun Classification from Predicate-Argument Structures", Proceedings of 28th Annual Meeting of the Association for Computational Linguist, 6-9 June 1990, Pittsburg, Pennysylvania, 268-275.

[139] Crouch, C.J. and Yang, B., (1992). "Experiments in Automatic Statistical Thesaurus Construction", Proceedings of the 15th Annual International ACM, SIGIR conference on Research and Development in Information Retrieval, 21-24 June 1992, Copenhagen, Denmark, 77-88.

[140] Grefenstette, G., (1994). Explorations in Automatic Thesaurus Discovery, Kluwer Academic Publishers, Boston, Dordrecht, London.

[141] Gasperin, C., Gamallo, P., Agustini, A., Lopes, G. and Lima, V. (2001). "Using Syntactic Contexts for Measuring Word Similarity", Workshop on Knowledge Acquisition and Categorization, ESSLLI.

[142] Curran, J. R. and Moens, M., (2002). "Improvements in Automatic Thesaurus Extraction", Proceedings of the ACL-02 Workshop on Unsupervised Lexical Ccquisition, ACL, 12 July 2002, Philadelphia, PA, USA, 59-66.

[143] Van der Plas, L. and Bouma, G., (2005). "Syntactic Contexts for Finding Semantically Related Words", Proceedings of the Meeting of Computational Linguistics in the Netherlands (CLIN), LOT Utrecht.

[144] Barzilay, R. and McKeown, K., (2001). "Extracting Paraphrases from a Parallel Corpus", Proceedings of the Association for Computational Linguistics, 2-7 June 2001, Pittsburgh, PA, USA, 50-57.

[145] Ibrahim, A., Katz, B. and Lin, J., (2003). "Extracting Structural Paraphrases from Aligned Monolingual Corpora", Proceedings of the 2nd International Workshop on Paraphrasing, Vol.16. Association for Computational Linguistics, 11 July, Sapporo, Japan, 57-64.

[146] Shimohata, M. and Sumita E., (2002). "Automatic Paraphrasing Based on Parallel Corpus for Normalization", Proceedings of the 3rd International Conference on Language Resources and Evaluation, 26-30 May 2002, Reykjavik, Iceland.

[147] Van der Plas, L. and Tiedemann, J., (2006). "Finding Synonyms Using Automatic Word Alignment and Measures of Distributional Similarity", Proceedings of 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, 17-21 July 2006, Sydney, Australia.

[148] Blondel, V.D. and Sennelart, P., (2002). "Automatic Extraction of Synonyms in a Dictionary", Proceedings of the SIAM Workshop on Text Mining, 13 April 2002, Arlington, VA, USA.

[149] Wang, T. and Hirst, G., (2012). "Exploring Patterns in Dictionary Definitions for Synonym Extraction", Natural Language Engineering, 18(3):313-342.

[150] Ehlert, B., (2003). Making Accurate Lexical Semantic Similarity Judgments Using Word-context Co-occurrence Statistics, Master's thesis, University of California.

[151] Freitag, D., Blume, M., Byrnes, J., Chow, E., Kapadia, S., Rohwer, R. and Wang, Z., (2005). "New Experiments in Distributional Representations of Synonymy", Proceedings of the 9th Conference on Computational Natural Language Learning, 29-30 June, Ann Arbor, Michigan, USA, 25-32.

[152] Jarmasz, M. and Szpakowicz, S., (2004). "Rogets Thesaurus and Semantic Similarity", Proceedings of Conference on Recent Advances in Natural Language Processing (RANLP), 10-12 September 2003, Borovets, Bulgaria, 212-219.

[153] Landauer, T. and Dumais, S., (1997). "Solution to Plato's Problem.The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge", Psychological Review, 104(2):211-240.

[154] Terra, E. and Clarke, C., (2003). "Frequency Estimates for Statistical Word Similarity Measures", Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, 27 May- 1 June 2003, Edmonton, Canada, 165-172.

[155] Turney, P.D., (2001). "Mining the Web for Synonyms: PMI–IR versus LSA on TOEFL", 491-502; Editors: Flach, P.A. and De Raedt, L. (2001). LNCS, 2167, Springer, Heidelberg.

[156] Turney, P.D., Littman, M.L., Bigham, J. And Shnayder, V., (2003). "Combining Independent Modules in Lexical Multiple-choice Problems", Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003, 101-110.

[157] Yates, A., Goharian, N. and Frieder, O., (2013). "Graded Relevance Ranking for Synonym Discovery", Proceedings of the 22nd International Conference on World Wide Web Companion, 13-17 May 2013, Rio De Janeiro, Brazil, 139-140.

[158] Hagiwara, M., Ogawa, Y. and Toyama, K., (2006). "Selection of Effective Contextual Information for Automatic Synonym Acquisition", Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, 17-21 July 2006, Sydney, Australia, 353-360.

APPENDIX-A

PATTERN SPECIFICATIONS

In Table A.1 shows the general patterns and Turkish equivalents. First column represents the general patterns which are widely used patterns in literature. Second column displays variation of Turkish equivalent of general patterns.

Table A.1 General Patterns and their Turkish equivalents

Dictionary-based Patterns	**Turkish Equivalent of General Patterns**
NPx is (a|-) part of NPy	…NPx…NPy+gen…(bir)? parçasıdır/kısımıdır …NPy+gen…parça/kısım(ları|sı)…NPx… …NPx…NPy+gen… parça/kısım(larından) biridir NPy+gen…parça/kısım(larından) biri olan…NPx
NPx member of NPy	…NPx…NPy+gen… (bir)? üyesidir …NPx… (bir)? NPy+nom…üyesidir …NPx…NPy+gen…üye(lerinden|sinden) biridir NPy+gen…üye(lerinden|sinden) biri olan…NPx
NPy constituted of NPx	…NPx…NPy+gen…bileşen(lerinden|inden) biridir NPy+gen…bileşen(lerinden|inden) biri olan...NPx …NPx…NPy+gen…(bir)? bileşenidir …NPy+gen…bileşen(leri|i)…NPx…
NPy made of NPx	NPy,…NPx+abl yapıl(mıştır|maktadır|ır) NPy,…NPx+abl yapılmış olup ...NPx+abl yapılan NPy
NPy consist of NPx	NPy,…NPx…içerir
has/have	NPy+gen...NPx+p3sg+nom...(vardır|var) NPx+p3sg+nom var olan NPy NPy+pnon+loc NPx (var|vardır)
with	NPx+p3sg+nom olan NPy

In Table A.2 shows the dictionary-based patterns and Turkish equivalents. First column represents the dictionary-based patterns which are obtained from dictionaries in

Turkish. Second column displays variation of Turkish equivalent of dictionary-based patterns.

Table A.2 Dictionary-based Patterns and their Turkish equivalents

Dictionary-Based Patterns	**Turkish Equivalent of Dictionary-based Patterns**
NPy,...(whole|group|all|set|flock|union) of NPx	NPy,…NPx+(gen|nom) bütünü(dür|-) NPy,…NPx+(gen|nom) topluluğu(dur|-) NPy,…NPx+(gen|nom) tümü(dür|-) NPy,…NPx+(gen|nom) birliği(dir|-) NPy,…NPx+(gen|nom) kümesi(dir|-) NPy,…NPx+(gen|nom) sürüsü(dür|-)
NPy, ...(class|member|team) of NPx	NPy,…NPx+(gen|nom) sınıfı(dır|-) NPy,…NPx+(gen|nom) üyesi(dir|-) NPy,…NPx+(gen|nom) takımı(dır|-)
NPx, ...family of NPy	NPx, NPy+gillerden NPy+gillerden…NPx
NPy, ...(amount|measure|unit) of NPx	NPy,…NPx+(gen|nom) miktarı(dır|-) NPy,…NPx+(gen|nom) ölçüsü(dür|-) NPy,…NPx+(gen|nom) birimi(dir|-)
NPy consist of NPx	NPx+abl oluş(an|muş) NPy NPy,...NPx+abl oluşmuştur
NPy made of NPx	NPx+abl...yapıl(an|mış) NPy
has/have	NPx+nom-adj-with NPy

APPENDIX-B

SYNONYM EXAMPLES

Table B.1 shows the examples of target words and its potentail synonym words in Turkish-English.

Table B.1 Target words and potential synonym words (nearest neighbors)

vasıta:vehicle	**kanıt:evidence**	**birey:individual**
gerekçe:justification	dosya:file	ilişki:relationship
çözgü:warp	loca:lodge	toplum:society
alternatif:alternative	veri:data	insan:man
rezonans:resonance	delil:evidence	sermaye:fund
taşıt:vehicle	ergene:-	performans:performance
dai:dai	beyanat:speech	üniversite:college
katil:murderer	örtbas:cover-up	yurttaş:citizen
mahremiyet:privacy	bulgu:discovery	tempo:pace
lakap:nickname	kulaklık:flap	vatandaş:citizen
çalıştay:workshop	iddianame:accusation	kişi:person
araç:means	ipucu:clue	
zabit:officer		
millet:people	**hata:fault**	**araba:car**
üreteç:generator	sorun:trouble	otel:hotel
bas:bass	zaman:date	yalı:waterside
hükümet:government	problem:problem	vagon:car
ülke:country	karar:judgement	villa:villa
herkes:everyone	sıkıntı: nuisance	kilise:church
parantez:parenthesis	yanlış:error	otobüs:coach
raslantı:coincidence		telefon:telephone
avrupa:europe		otomobil:automobile
ulus:nation		cep:pocket
halk:community		ev:house
toplum:society		okul:school

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname : Tuğba YILDIZ

Date of birth and place : 15.03.1980 / İSTANBUL

Foreign Languages : English

E-mail : tuubayildiz@gmail.com

EDUCATION

Degree	Department	University	Date of Graduation
Master	Computer Engineering	Kocaeli University	2006
Undergraduate	Computer and Mathematics	İstanbul Bilgi University	2003
	Business Administration (Minor Program)	İstanbul Bilgi University	2003
High School	Kabataş Erkek Lisesi		1998

WORK EXPERIENCE:

Year	Corporation/Institute	Enrollment
2003 - 2007	İstanbul Bilgi University	Teaching Assistant
2007 - …	İstanbul Bilgi University	Teaching Staff

PUBLISHMENTS

Papers

1. Yıldız, T., Diri, B., Yıldırım, S., (2014). "Acquition of Turkish Meroym based on Classification of Patterns", Knowledge Based Systems, Elsiever (SUBMITTED)

2. Yıldız, T., Diri, B., Yıldırım, S., (2014). "An Approach to Turkish Synonym Extraction from Multiple Resources: Monolingual Corpus, Mono/Bilingual Dictionaries and WordNet", Computer Speech and Language, Elsiever (SUBMITTED)

Conference Papers

1. Yıldız, T., Diri, B., Yıldırım, S., (2014). "Analysis of Lexico-Syntactic Patterns for Meronym Extraction from a Turkish Corpus", 6th Language and Technology Conference, Human Language Technologies as a Challenge for Computer Science and Linguistics, LTC, Poland.

2. Yıldız, T., Yıldırım, S., (2012). "Association Rule Based Acquisition of Hyponym and Hypernym Relation from a Turkish Corpus", Innovations in Intelligent Systems and Applications (INISTA), 2012 Trabzon/Turkey

Books

1. Yıldırım, S., Yıldız, T., (2012). "Corpus-Driven Hyponym Acquisition for Turkish Language", 29-41; Editor: Gelbukh, A., CICLing 13th International Conference on Intelligent Text Processing and Computational Linguistics 2012, LNCS 7181, Springer Berlin / Heidelberg.

2. Yıldız, T., Yıldırım, S., Diri, B., (2013). "Extraction of Part-Whole Relations from Turkish Corpora", 126-138; Editor: Gelbukh, A., CICLing 14th International Conference on Intelligent Text Processing and Computational Linguistics 2013, LNCS 7816, Springer Berlin / Heidelberg.

3. Yıldız, T., Diri, B., Yıldırım, S., (2014). "An Integrated Approach to Automatic Synonym Detection in Turkish Corpus", 116-127; Editors: Przepirkowski, A. and Ogrodniczuk, M., 9th International Conference on Natural language Processing, POLTAL 2014, LNAI 8686, Springer International Publishing Switzerland.

AWARDS

1. Best Paper Award: Second Place

Yıldırım, S., Yıldız, T., (2012). "Corpus-Driven Hyponym Acquisition for Turkish Language", CICLing 13th International Conference on Intelligent Text Processing and Computational Linguistics 2012, Proceedings Part I, LNCS 7181, Springer Berlin / Heidelberg, 29-41