AN APPROACH FOR DICTIONARY-BASED CONCEPT MINING IN TURKISH

by

Cem Rıfkı Aydın

B.S., Computer Engineering, Bahçeşehir University, 2011

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2014

AN APPROACH FOR DICTIONARY-BASED CONCEPT MINING IN TURKISH

APPROVED BY:

Assoc. Prof. Tunga Güngör
(Thesis Supervisor)

Assist. Prof. Günizi Kartal

Assist. Prof. Arzucan Özgür

DATE OF APPROVAL: 20.09.2013

ACKNOWLEDGEMENTS

To my family.

I am grateful to my thesis advisor, Assoc. Prof. Tunga Güngör, for his persistent help and guidance throughout my thesis work. His immense knowledge regarding the field of my thesis work, and that he made quite meaningful proposals on how to ameliorate my thesis algorithms and studies made me get more motivated.

This work was supported by the Boğaziçi University Research Fund under the grant number 5187, and the TÜBİTAK under the grant number 110E162. I am grateful to TÜBİTAK (Scientific and Technological Research Council of Turkey) for their financial support, and also for that I was awarded scholarship throughout my M.Sc. education, under the grant number 2210. That also made me get more motivated while preparing my thesis.

I am grateful to Assist. Prof. Arzucan Özgür, and Assist. Prof. Günizi Kartal for that they accepted to participate in my thesis committee.

I am grateful to Ali Erkan, with whom I worked for a TÜBİTAK sponsored project, topic of which is same as my thesis work. Through the brainstorming, I came up with more creative solutions and proposals, and this made me more enthusiastic on my studies.

I am grateful to my mother, father, and sister for their consistent love they have had for me since I was born. Their support throughout my life made me be more powerful in terms of both spirit, and success in my academic studies as well as life itself. They mean everything to me.

ABSTRACT

AN APPROACH FOR DICTIONARY-BASED CONCEPT MINING IN TURKISH

Concept mining is a field of natural language processing, where the documents that may be text files, e-mails, papers, journals, or any other textual materials are scanned, and comprehensive concepts concerned with these documents are created. Here, concepts can be thought of as general ideas extracted from the documents. Concepts can also be extracted from visual and audio materials, nonetheless this thesis focuses on extracting concepts from only textual materials, in an efficient way in terms of time, quality and accuracy. In NLP field, the difference between keyword and concept should be noticed such that keyword has to explicitly occur in the material being scanned, whereas concepts do not have to appear in these materials. This is quite a big challenge, which may call for the use of NLP or statistical methods, which may be beneficial for extracting expressive concepts. So far, numerous studies have been performed especially in western languages, such as English, French, German and Spanish amongst many, and quite successful results have been achieved. As for Turkish, this topic is still quite immature compared with the languages mentioned above. It has to be taken into consideration that Turkish is an agglutinative language, therefore the documents first need to be pre-processed in order to get word stems. Among these words, we take only nouns into account, since concepts are generally considered noun. This thesis utilizes statistical methods, and the official Turkish dictionary. The statistical method counts the frequency of words, whereas the use of dictionary may suggest some probable concept words that do not appear in the documents. The success rate (precision) for this concept extraction method is 63.97%.

ÖZET

TÜRKÇE İÇİN SÖZLÜK TABANLI BİR KAVRAM ÇIKARMA SİSTEMİ GELİŞTİRİLMESİ

Kavram madenciliği, basit metin dosyalarının, elektronik postaların, akademik yazıların, gazete kupürlerinin veya başka metin materyallerinin taranıp, bu dokümanlardan en kapsamlı kavramların belirlendiği, Doğal Dil İşlemenin bir alanıdır. Burada kavramlar dokümanlardan çıkarılmış genel fikirler olarak düşünülebilir. Kavramlar aynı zamanda görsel veya işitsel materyallerden de çıkarılabilir; ama bu tez, zaman, kalite ve doğruluk açısından verimliliği amaç edinerek, sadece metinsel dokümanlardan kavram çıkarma üzerine odaklanmıştır. Doğal Dil İşleme alanında anahtar kelime ile kavram arasındaki fark, anahtar kelimenin dokümanda geçebilirken, kavramların dokümanda geçme zorunluluğu olmamasıdır. Bu, anlamlı kavramlar çıkarılabilmesine olanak sağlayan Doğal Dil İşleme ve istatistiksel metotların kullanılmasını gerekli kılabilir. Bu alan, İngilizce, Fransızca, Almanca, İspanyolca ve diğer birçok Batı dillerinde üzerinde çalışılmakta ve çok başarılı sonuçlar elde edilmektedir. Türkçede ise bu konu üzerine diğer dillere kıyasla çok çalışma olmamıştır. Türkçe sondan eklemeli bir dildir, bu yüzden dokümanlar önce bazı işlemlerden geçirilmeli, sonra da kelimelerin kökleri işleme tabii tutulmalıdır. Bu kelimeler arasından sadece isimler göz önünde bulundurulmalıdır; çünkü kavramlar genelde isimler olarak düşünülmektedir. Bu tez çalışmasında istatistiksel metot ve Türkçe sözlüğünden yararlanılmıştır. İstatistiksel metot kelimelerin bulunma sıklığını hesaba katan bir yol izlerken, sözlük kullanımı da dokümanda yer almayan kelimeleri olası kavram olarak önerebilmektedir. Bu tez kavram çıkarma metodunun başarı oranı yüzde 63.97 olarak belirlenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

AI	Artificial Intelligence
BoDis	Boun Morphological Disambiguator
BoMorP	Boun Morphological Parser
CM	Concept Mining
HMM	Hidden Markov Model
IDF	Inverse Document Frequency
IR	Information Retrieval
LDA	Latent Dirichlet Allocation
LM	Language Model
LSA	Latent Semantic Analysis
ML	Machine Learning
NLP	Natural Language Processing
POS	Part-of-Speech
SVM	Support Vector Machine
TDK	Türk Dil Kurumu
TF	Term Frequency
UTF-8	8-bit Unicode Transformation Format
XML	Extensible Markup Language

1. INTRODUCTION

In the recent decades, a great amount of materials have been produced, especially in electronic format, many of which are processed and used in accordance with the need of people. For example, search engines may index the web site contents, and after processing those sites, people may benefit from these materials with regard to their interests. The most common approach, in search engine domain, to get the relevant page to the needs of the user is to make use of the queries being composed of keywords. These keywords generally have to occur in the documents if some of them are to be returned, and some specific AI algorithms may be applied to measure the relevance between the keywords typed in and the documents. Besides search engines, one may want to get a general information about a web site, blog, e-mail, survey, video or audio file, database, or some any other material. It is the case that the users do not have to know the keywords, according to which some documents are to be returned or processed, so some generalized knowledge concerning those documents may be extracted, and the users may have a general idea about them.

Concept is a term used in many contexts, nonetheless its main definition is given in the domain of ontology, a field of study in philosophy. In this regard, concepts can be thought of as mental representations of objects, abstract objects and constituents of propositions which make them mediate between language and thought, or abilities that are peculiar to agents [1]. In this aspect, concepts can be thought of as generalized representations of words, which are at a level of higher abstraction. For example the word *organism* may be a probable concept candidate for the word *animal* since the former word is at a higher level of abstraction compared with the latter one. An abstract, or concrete object representing a word may have one or many concepts, also a concept may correspond to many words.

Concept extraction can be performed in two ways as follows:

(i) Expert-based approach,
(ii) NLP or Statistical approach.

In the expert-based approach, the documents can be examined by the humans, who read through the documents and extract concepts from them. It has many advantages, but it may be time consuming, and financial problems may be a challenge. Instead, the second method that include NLP and statistical approaches can be used. NLP and statistical methods implement some AI algorithms that can be applied to extract concepts, some of which are clustering, latent semantic analysis (LSA), hidden Markov model (HMM) and support vector machines (SVM) amongst many others. The difference between statistical and NLP approaches is that human intervention is possible in the latter one [2]. These approaches may be economic and time efficient, nevertheless the accuracy may be not as high as achieved by expert humans.

The majority of studies in the field of concept mining is carried out for English, and many commercial software applications are built for this purpose. Some of such software applications are AlchemyAPI, WordStat and SPSS PASW Text Analytics. The first one extracts concepts only in English, but it also produces keywords, other categorical as well as semantic attributes concerned with the documents it processes and performs sentimental analysis, whereas the latter two software applications provide concept extraction applications in many languages besides English, such as French, German, Arabic, Spanish, and some others. When it comes to Turkish, there is not such a known concept extraction software developed for this language, whereas there are some software applications, which can extract keywords or key phrases as in [3, 4]. The difference is that keywords or key phrases have to explicitly occur in the documents being scanned whereas concepts may not be obligated to appear in the document. This thesis proposes a new method for extracting concepts from Turkish documents through a new algorithm. So far, even in English the use of dictionary apart from WordNet has rarely been encountered, therefore the use of dictionary in this thesis may be considered a novel approach.

The outline of this thesis work is as follows: Chapter 2 is concerning the literature survey, and related works are mentioned. In Chapter 3, the novel algorithm that is developed for this thesis work is explained in detail. Chapter 4 shows the experiments, evaluations, and results produced. Finally, Chapter 5 concludes the paper.

2. LITERATURE SURVEY

Concept mining is a field where many studies are performed, and it has an increasing importance due to that massive amounts of electronic materials are needed to be processed. For example, when searching something through a search engine or when one looks at the document, the users may not want to read all through the material, instead they may want to look at keywords, to decide whether this document is relevant to what they are searching for in a short period of time. Whereas the keywords have to explicitly occur in the documents, the situation for concepts are different: A concept may appear or not in the document being processed, and a concept often represent more generalized abstract ideas. Giving concepts of a document might also make the reader decide whether the document is relevant to his/her inquiry and have a general knowledge concerning this document before even starting to read it.

Concept extraction is used in not only the field of information retrieval (IR), but also in many different fields of studies and sectors. Some of the domains, where concept mining is utilized besides IR, can be listed follows:

- Medical use as in [5, 6]. Detection of cancer areas can be an example of extraction of concepts from visual material. Detecting the most common diseases in a specific patient population in this respect can also be thought of as another example of concept mining.
- Legal cases [7]. Categorizing the judiciary classes, such as adult court, appellate court and many others are some examples of concept extraction in this domain.
- Banking systems. Banks may track the profiles of the creditworthy customers and make offers, this can also be considered a concept mining method, however, privacy violation can be the matter here. Fraud detection is another example of this field as well.
- Satellite images can be arranged, and identified (such as urban or rural areas) with concept mining method [5].

- Results of surveys, that are open-ended, can be evaluated through the use of Concept Mining methods.

Concept mining has a wide range of utility, but most of the studies are based on extracting concepts from textual materials, whereas there are not numerous studies carried out for extracting concepts from audio or video materials.

2.1. Some Methods Examined in Concept Mining Field

In the field of studies concerned with concept mining, generally AI algorithms as well as different dictionaries and lexical databases are utilized. Some papers propose a method which makes use of statistical methods, whereas some others make use of NLP algorithms. The most widely used lexical database is WordNet in this field, because it has a unit called synset, which determines the relationship between words, taking into account that the relations between words may help semantic relevance be shown and expressive concepts be extracted. Some papers propose the use of clustering, a machine learning (ML) algorithm, whereas some others make use of latent Dirichlet allocation (LDA), HMM and many other methods.

Initially, as this thesis is concerned with concept mining in Turkish, the algorithm stated in paper by Meryem Uzun-Per [8], which is also regarding concept mining in the same language, is carefully examined. In this paper, k-means clustering method, being an AI method, is utilized. First, documents are parsed, and thereafter are disambiguated in order to get the word stems eliminating inflectional morphemes, taking into account that Turkish is an agglutinative language. Then only nouns are taken into consideration as concepts are generally considered nouns. But this thesis work does not offer a thoroughly automatic method, it also counts on the human-specialist's contribution. First document-noun matrix is built that shows the frequencies of column representative nouns in the row representative document nouns. Then, consistent with this matrix, clusters are constructed including those nouns. Those clusters afterwards are assigned to documents according to a threshold value. A ratio that takes account of the division of the frequency of nouns in a document, which are also in specific cluster, by the total number of words in that cluster is

tested against that threshold value. If the ratio exceeds that ratio, then cluster might be assigned to that document. Then, through the help of human specialist, concepts are assigned to those clusters, and then those concepts are indirectly assigned to the documents. In this study, also key files are created for each separate document, and these are used in the testing phase. The success rate produced in this work is 51%.

The algorithm proposed by Elberrichi *et al.* [9] utilizes the lexical database WordNet, which has relation sets called synset. Synsets are composed of many relations such as hypernymy, hyponymy, synonymy, and many others. Here hypernymy corresponds to a relation, according to which one word is a more general form of the other one. For example, the word *animal* is a hypernym of the word *cat*, and it is not a symmetric relation. The important point here to note is that the selected relationship as input for algorithm is this relation, being hypernymy, since concepts represent also, like hypernyms, general forms (ideas) of other words. Initially stop words are eliminated, such as *the* and *an*, hereafter noun phrases are taken into consideration. This can be considered a good approach since in many studies only nouns separately are taken account of, not noun phrases. So it can be said that this work is not based on a bag-of-words model. According to the algorithm, frequencies are taken into account. All hypernyms of words are taken, and they are valued with the frequencies of these words. Then, whichever hypernym word has the utmost score assigned, it is declared as the probable concept of the document. For example, if there are words in the document such as *football*, *handball* and *attorney*, and their frequencies are two, one, and two respectively, the hypernyms would be *'sport*', '*sport*', and '*law*' respectively, and the values for those hypernym words would be again two, one, and two. The hypernym word *sport* is seen twice, so its frequencies should be summed up, that is, it must be 2 + 1 = 3, whereas the hypernym *law* should have a value of 2. Therefore the concept for this document would be *sport*. Consistent with this study, this algorithm is combined with another one, that is text categorization, and success rate is reported to be 71%.

Another study on this field is performed by Liu, and Singh [10]. In this paper, ConceptNet, a freely available large-scale common-sense knowledge database is explained. It is similar to the lexical database WordNet such that words in ConceptNet are connected to each other in accordance with their semantic relevances as words in the latter

one are also connected to each other through the relations called synsets. The difference is that ConceptNet is much more comprehensive than WordNet. ConceptNet can be thought of as a concept mapping that links nodes, which are word phrases that may be verbs, nouns, or other word groups, through semantic relationships. For example, the property *IsA* in this graph can be thought of as the hypernymy relation, but the properties such as *PropertyOf*, *MotivationOf* (affect), *CapableOf* (agent's ability), and many others cannot be found as some synset relations in WordNet. Therefore the use of this knowledge base may be beneficial. The relations in this graph based knowledge base can also be extended. For example if there is a relation such as *<IsA 'apple' 'fruit'>*, and *<PropertyOf 'apple' 'sweet'>*, then a new relation would be implied, such as *<PropertyOf 'fruit' 'sweet'>*. These new extended relations may help concept extraction system achieve higher accuracy results. When a document is sent as input for concept extraction, first, the concept mapping is created. This graph can be thought of as nodes representing the word phrases in the documents, and edges linking them according to their relational properties. If some of the nodes in the graph have many links as input and output, the words representing those nodes may be labeled as probable concepts. This is meaningful since this relevance between words may show the semantic relationship between them, and more links around a node show that a specific word phrase has relevance to many other nodes in the graph, which makes that word a candidate for a general word, that is concept, in the context of the document. This knowledge base is developed in English, and there is no support for other languages.

In the study performed by Ramirez *et al*. [11], a concept extraction method is developed for web sites. In accordance with this algorithm, first, web pages are parsed since these pages have many tags such as *<html>*, *<body>*, *<title>*, and others, which may not contribute to the set of probable concepts. Then stop word elimination is performed, and words are added to the concept set according to their frequencies. If the frequency of a word exceeds a specific threshold value, it is added to the concept set. It is meaningful, since general idea of a document is generally related to the most frequent words in the document or words relevant to that most frequent words. Then the approach used takes account of html tags, only eliminating some specific tags such as *javascript*, *style*, and some others. Each word group residing between tags is given a weight score. For example, the words between the tag *<title>* or *<b>* are assigned higher scores. After

scoring operation, if also this score exceeds the threshold value, the word groups are added to the concept set. This is reasonable, since words between some tags have a higher importance compared with other words between other tags. Also noun phrases are taken into account in this study, which makes this method be a non bag-of-words approach. Accuracy results achieved for this study are reported to be high.

Finally, a paper shows a novel approach that is developed for concept mining domain [12]. This study is concerning topic digital library construction, and concepts are extracted from documents, accordingly documents are categorized via clustering. In order to extract concepts, method as follows is implemented: At first, an equation is created which takes into account many factors concerning a term, and the multiplication of those factors produces a score. The factors are term frequency (TF), inverse document frequency (IDF), position of the first occurrence, and distribution deviation of the keywords. Here, whichever words give the highest scores, they are selected as probable concepts for the document being processed. Then, through the concepts gathered as explained above, a concept matrix for documents is built. Afterwards k-means algorithm is implemented to cluster the documents consistently with these concepts. The success results are reported to be high.

2.2. Some Popular Software Developed for Concept Mining

Although many studies have been carried out concerning concept mining in NLP field, there are not many software applications that are popular and widely used for it. Some of the reasons for it can be that most of these applications are commercial, and Concept Mining is still an area that is not well-known by people, or people do not know how they will benefit from it. Nonetheless as for companies, there are some widely used commercial software, with the most popular and widely used ones being SPSS Inc., WordStat, and a relatively new software AlchemyAPI. The first software tool provides concept mining functions for many languages, such as in English, French, German, Spanish, Arabic, and many others, whereas the second one works for English, French, Italian, and German, and the last one offers concept extraction just for English. But AlchemyAPI provides other functionalities besides extraction of concepts in some other languages, such as sentiment analysis in English and German, whereas entity extraction is

provided in eight languages that are English, German, French, Italian, Spanish, Portuguese, Swedish, and Russian.

These software tools are used in textual Concept Mining, and offer many utilities. Some of them are fraud detection, keyword extraction, analysis of surveys which are open-ended, document classification and extracting information from reports among many others. These tools provide graphically advanced visualization techniques as well as tables to show the concepts, their relevance, and their relations.

3. METHODOLOGY

In this work, four corpora are processed, which are collected by Gazi University. These corpora are pre-processed to extract the nouns, because the concepts are generally thought of as nouns as mentioned. Pre-processing is performed by the parser, and disambiguator tools developed by Hasim Sak, at Boğaziçi University. Afterwards, the nouns are used in the method in accordance with this study, and expressive concepts are extracted.

In this study, algorithm developed is implemented on four corpora, all of which are collected from sources in Turkish, and they are in *.txt* format. These corpora are as follows:

(i) Sports News Corpus: This corpus has documents that are concerned with sports news collected from Turkish sources. The majority of news is regarding football. The major topic is about the results and scores of matches between different teams. Remarks by sports team players are also encountered in this corpus. This corpus has 100 documents, length of each of which is, on average, not large.

(ii) Forensic News Corpus: This corpus has documents that are concerned with news in the field of forensic subject from Turkish sources. The majority of news is concerning the events being crime or abuse incidents, and decisions made by judges. This corpus has also 100 documents, with their length being not large.

(iii) Forensic (Court of Appeals Decisions) Corpus: This corpus has documents that are concerned with court of appeals decisions. It is similar to the Forensic News corpus, however it is more comprehensive. The documents of corpus is collected from different Turkish forensic sources, and the prevalent topic is regarding the crimes, or abuses, and the decisions made by the judges. This corpus has 108 documents, which makes it the largest corpus in terms of number of documents, also the length of documents on average is not large.

(iv) Gazi Corpus: This corpus has documents that are concerned with different fields of engineering. For example, some of the documents are regarding electrical engineering information, some are concerning architectural reports, and some are regarding civil engineering amongst many others. The distribution of topics over

different engineering topics is homogeneous. This corpus has 60 documents, making it the smallest among corpora in terms of number of documents. But the length of each file, on average, is large.

In this work, the concepts are extracted from each of the files in these corpora, and it is noticed that the files that are in the same corpus have similar concepts, with the exception being Gazi Corpus, such that it is more heterogeneous in terms of topics it has, compared with the three others.

3.1. Pre-processing on Files in Corpora

The files to be processed are in Turkish, therefore the Turkish characters needed to be taken into account. In order to process these characters, UTF-8 format has to be used in files.

UTF-8 is a format, according to which variable-width encoding, that can represent any character in the Unicode character set, is used. It is the most widely used character encoding in World Wide Web, also its popularity as the default encoding system in operating systems, software applications, and programming languages is increasing as compared with other formats.

UTF-8 encodes Unicode characters in a way using one to four 8-bit bytes, which are called *octets* in the Unicode standard. It encodes the characters having lower values with fewer bytes, which are in earlier positions in Unicode character set and occur more frequently.

First, the tokenization process should take place. In accordance with this process, the punctuation characters are separated from other characters by a blank space to right, and to left. For example, the sentence below is the definition of the word *trough* in English dictionary:

"A long, narrow, generally shallow receptacle for holding water or feed for animals."

The output of this sentence, after tokenization process is implemented, can be as follows:

"A long , narrow , generally shallow receptacle for holding water or feed for animals ."

As can be noted, the difference between two sentences is that there are some extra blank space characters before and after punctuation characters, which in this case are comma and period characters. If there have already been white spaces preceding punctuation marks, then such a processing operation would not be needed.

3.2. Parsing and Disambiguation Processes

The files in the corpora that are to be processed are in an unstructured form as most of the textual files in electronic format are. In order to extract concepts, nouns in the documents have to be gathered and listed, hence stemming operations that eliminate the suffixes are necessary. In English, there are not many inflectional and derivational suffixes, so suffix elimination might not be quite a challenge, nevertheless as for Turkish, the situation is different in that it is an agglutinative language. Many inflectional and derivational suffixes lead to complexity when performing stemming operation.

We process the words after they are parsed into morphemes that may be derivational or inflectional, because we need to get the stems of the words. Among these stems, we pick up only the ones being nouns. We have to eliminate the words that are of other categories, such as adjectives, verbs or nouns.

Although, in the field of Concept Mining, the majority thinks that the concepts should be nouns, some people think [10] they can be verb as well. It is reasonable, since a verb has a definitive effect on the meaning of a sentence, and it indirectly affects the semantics of the whole document. For example, if a document has many verbs such as *beat*, some of the probable concepts of this document would be *win* or *victory*. But in this study, the majority unanimity is accepted: Concepts should be thought of as being nouns.

In order to extract the nouns from files in the corpora, some parsing and disambiguation tools are needed to be used, therefore the Boun Morpohological Parser (BoMorP) and the Boun Morphological Disambiguator (BoDis) tools [13, 14] are utilized. These tools are developed at Boğaziçi University.

The parser simply parses the words in the document, separates and shows the inflectional and derivational morphemes. In order to achieve successful results, the above-mentioned tokenization process needs to be implemented, because although in English, there are some words having both punctuation and alphabetic characters such as the word "can't", there is not such a known word in Turkish according to which, when punctuation character in this word is removed, remaining words are both nouns. If we do not eliminate the punctuation characters, we may encounter a lower success rate. For example, a word in the document can be as follows:

"ölümsüzleştiriveremeyebileceklerimizdenmişsinizcesine"

The above word can be broken into its morphemes by a parser as follows:

"öl + üm + süz + leş + tir + iver + e + me + yebil + ecek + ler + imiz + den + miş + siniz + ce + sin + e"

The example above shows the derivational and inflectional richness of Turkish, an agglutinative language. When we try to parse languages such as English, French, Italian, Spanish or Portuguese, we may not encounter such morpheme richness due to that these languages family are inflectional. So developing a parser for such agglutinative languages, such as Turkic languages, Finnish, Hungarian and Estonian, requires much more effort.

This parser is a finite-state machine, which is composed of three components:

(i) A lexicon that contains the stems of the words in Turkish. This is needed since the roots can only then be found and used.
(ii) A morphotactics component (morphosyntax) that defines the ordering of the morphemes.

(iii) A morphophonemics component that determines the phonetic variations when morphemes are added during the word formation.

Also there is a fact that parser may return many possible parsing suggestions, for example the word *çekin* may be parsed, and the following outputs can be encountered:

çekin[Verb]+[Pos]+[Imp]+[A2sg]
çeki[Noun]+[A3sg]+Hn[P2sg]+[Nom] (3.1)
Çekin[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom]

Above, there are some parsed forms of the word *çekin*. In the first one, it is simply a verb in imperative mood, in the second one, it is a noun in possessive form, whereas the last one is a proper noun. The abbreviation *A3sg* stands for the third singular person inflection, whereas *P2sg* stands for second plural inflection. But one cannot be sure which one of the above forms are used in the context of the word in the document by using only the parser tool. Therefore, a scoring process must be performed and one of the parsing outputs should be returned that has the highest score. We need to use disambiguator tool for it.

Disambiguator tool takes parsed files as input and disambiguates the words, that is, it selects the most accurate parsed alternative taking into account the context. In order to disambiguate the parsed words, an averaged perceptron-based algorithm is utilized. In order to select the most accurate alternative, a scoring mechanism is used, and this tool gives a success rate of over 97%, which has been the highest one achieved in Turkish so far. Table 3.1 gives an example of the parsing output of the below sentence, present in Forensic News corpus, using BoMorP, whereas Table 3.2 gives the disambiguation results for this sentence, taking the output from the parser as an input. It can be clearly seen that scores are taken into account to determine the best-matching disambiguated word. In Table 3.2 it is assumed that Part-of-Speech (POS) tags are lined up in a decreasing order in terms of score.

"Mahkeme Başkanı Alçık, sanık isimlerini tek tek okudu sanıklar ise el kaldırarak savunması yapıldı."

Table 3.1. An example of parsed output.

Mahkeme mahkeme[Noun]+[A3sg]+[Pnon]+[Nom]
Başkanı
başkan[Noun]+[A3sg]+[Pnon]+YH[Acc]
başkan[Noun]+[A3sg]+SH[P3sg]+[Nom]
Başkan[Noun]+[Prop]+[A3sg]+SH[P3sg]+[Nom]
başka[Adj]-[Noun]+[A3sg]+Hn[P2sg]+NH[Acc]
Alçık Alçık[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom]
,
,[Punc]
sanık
sanık[Adj] sanık[Noun]+[A3sg]+[Pnon]+[Nom]
isimlerini
isim[Noun]+[A3sg]+lArH[P3pl]+NH[Acc]
isim[Noun]+lAr[A3pl]+SH[P3sg]+NH[Acc]
isim[Noun]+lAr[A3pl]+SH[P3pl]+NH[Acc]
isim[Noun]+lAr[A3pl]+Hn[P2sg]+NH[Acc]
tek
tek[Adj]
tek[Noun]+[A3sg]+[Pnon]+[Nom]
TEK[Noun]+[Acro]+[A3sg]+[Pnon]+[Nom]
Tek[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom] tek[Adv]
tek
tek[Adj]
tek[Noun]+[A3sg]+[Pnon]+[Nom]

Table 3.1. An example of parsed output (cont.).

```
TEK[Noun]+[Acro]+[A3sg]+[Pnon]+[Nom]
Tek[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom] tek[Adv]

okudu
oku[Verb]+[Pos]+DH[Past]+[A3sg]

sanıklar
sanık[Adj]-[Noun]+lAr[A3pl]+[Pnon]+[Nom]
sanık[Noun]+lAr[A3pl]+[Pnon]+[Nom]

ise
i[Verb]+[Pos]+sA[Cond]+[A3sg] is[Noun]+[A3sg]+[Pnon]+YA[Dat]

el
el[Noun]+[A3sg]+[Pnon]+[Nom]

kaldırarak
kal[Verb]-DHr[Verb+Caus]+[Pos]-YArAk[Adv+ByDoingSo]
kaldır[Verb]+[Pos]-YArAk[Adv+ByDoingSo]

savunması
savun[Verb]+[Pos]-mA[Noun+Inf2]+[A3sg]+SH[P3sg]+[Nom]

yapıldı
yap[Verb]-Hl[Verb+Pass]+[Pos]+DH[Past]+[A3sg]

.
.[Punc]
```

Output of the disambiguator program taking the above parsed file as input is as follows:

Table 3.2. An example of disambiguated output.

Mahkeme
mahkeme[Noun]+[A3sg]+[Pnon]+[Nom]

Başkanı
başkan[Noun]+[A3sg]+SH[P3sg]+[Nom]
başkan[Noun]+[A3sg]+[Pnon]+YH[Acc]
Başkan[Noun]+[Prop]+[A3sg]+SH[P3sg]+[Nom]
başka[Adj]-[Noun]+[A3sg]+Hn[P2sg]+NH[Acc]

Alçık
Alçık[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom]

,
,[Punc]

sanık
sanık[Noun]+[A3sg]+[Pnon]+[Nom]
sanık[Adj]

isimlerini
isim[Noun]+lAr[A3pl]+SH[P3sg]+NH[Acc]
isim[Noun]+[A3sg]+lArH[P3pl]+NH[Acc]
isim[Noun]+lAr[A3pl]+SH[P3pl]+NH[Acc]
isim[Noun]+lAr[A3pl]+Hn[P2sg]+NH[Acc]

tek
tek[Adj] tek[Noun]+[A3sg]+[Pnon]+[Nom]

Table 3.2. An example of disambiguated output (cont.).

TEK[Noun]+[Acro]+[A3sg]+[Pnon]+[Nom]
Tek[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom]
tek[Adv]

tek
tek[Adj] tek[Noun]+[A3sg]+[Pnon]+[Nom]
TEK[Noun]+[Acro]+[A3sg]+[Pnon]+[Nom]
Tek[Noun]+[Prop]+[A3sg]+[Pnon]+[Nom]
tek[Adv]

okudu
oku[Verb]+[Pos]+DH[Past]+[A3sg]

sanıklar
sanık[Noun]+lAr[A3pl]+[Pnon]+[Nom]
sanık[Adj]-[Noun]+lAr[A3pl]+[Pnon]+[Nom]

ise
i[Verb]+[Pos]+sA[Cond]+[A3sg]
is[Noun]+[A3sg]+[Pnon]+YA[Dat]

el
el[Noun]+[A3sg]+[Pnon]+[Nom]

kaldırarak
kaldır[Verb]+[Pos]-YArAk[Adv+ByDoingSo]
kal[Verb]-DHr[Verb+Caus]+[Pos]-YArAk[Adv+ByDoingSo]

savunması
savun[Verb]+[Pos]-mA[Noun+Inf2]+[A3sg]+SH[P3sg]+[Nom]

Table 3.2. An example of disambiguated output (cont.).

yapıldı yap[Verb]-Hl[Verb+Pass]+[Pos]+DH[Past]+[A3sg]   . .[Punc]

The important point to note here is that these parser and disambiguator tools also can identify numbers, and punctuations. It is useful since these characters may be for some algorithms, and many parser, and disambiguator tools developed for many languages generally overlook these types of characters.

If we look at the words shown in Table 3.1 and Table 3.2, it can be seen that there are also sub-types for nouns. For example, a word can be a proper noun and we have to eliminate this alternative, because a proper noun such as *Christina* cannot be an abstract, general idea of a document. We also have to eliminate the abbreviation and acronym nouns since these can't represent concepts. For example, the noun *m* may stand for the noun *meter* as abbreviation, or *UN* may stand for *United Nations*, they may not help us determine the general concepts of a document, so those types of nouns should also be eliminated.

3.3. Previous Algorithms Developed that do not use Dictionary

There have initially been developed some algorithms for this study, but it is seen that they could not produce meaningful results. Therefore, new algorithms that make use of dictionary are developed, as will be explained in Section 3.4. The previous algorithms that were developed are as follows:

- Sentence Co-occurrence Algorithm: In accordance with this algorithm, the sentences in corpora are thought to represent semantic relationships between words. If a couple of words co-occur in many sentences, it would mean that those words are semantically related to each other. In order to extract this relationship, a square

matrix is built that stores scores indicating in how many sentences two words co-occur. Row and column words are assumed to be same in this matrix. For example (*i, j*)th element of matrix indicate in how many sentences *i*th and *j*th words co-occur in the corpus. The diagonal elements are updated as the frequencies of those words in the corpus, since diagonal elements represent the row and columns corresponding to the same word. Then the matrix is normalized by dividing all the row elements' values by the corresponding diagonal row element to get more sensible results. Finally, clustering methods are implemented, which are k-means, c-means and hierarchical clustering. K-means algorithm initially chooses k random points on an n-dimensional plane and through iterations, these points' feature values are recalculated as the mean values of the data features that are closest to those points, until convergence is met [17]. While in k-means algorithm, a sample can belong to only one cluster, in c-means it is also possible for a sample to belong to more than one cluster. Lastly, hierarchical clustering simply puts the nearest samples in one cluster, then expand this cluster's range by adding another nearest samples into itself, until all samples are assigned to a cluster. This is called agglomerative clustering that is implemented for this algorithm. But three clustering methods that we implemented produced unsuccessful results. The clusters created had words that are irrelevant to each other within, therefore the sentence co-occurrence method had to be dismissed.

- Window Co-occurrence Algorithm: After seeing the unsuccessful clusters constructed by sentence co-occurrence algorithm, another approach is developed. In accordance with this algorithm, windows are used in order to extract semantic relationships between words. Windows are simply the word groups in which words come one after another in a specified window size. The most commonly used window sizes are 30, 50, 70 and 100, and all these sizes are taken into account for this study. These windows are sliding ones, that is, after one iteration the starting position of one window is shifted one word rightwards. Also, each word that co-occurs in one window is not assumed to be co-occurring in the very next sliding window one more time. A square matrix, as is the case for sentence co-occurrence algorithm, is built. The values in this matrix are filled in accordance with the number of windows in which two words co-occur. Diagonal elements are updated as the corresponding row (or column) word's occurrence frequency in windows. Then again, k-means, c-means and hierarchical clustering methods were implemented. For

all these clustering methods, there were only a few clusters that had words relevant to each other within, so this algorithm had to be dismissed as well.

- Dictionary Clustering Algorithm: After two algorithms mentioned above produced unsuccessful results, another approach is developed, taking dictionary structure into account. In dictionaries, definitions of word entries may show the semantic relationships between words, as will be explained in detail in Section 3.4. Consistent with this algorithm, corpus nouns are taken account of, and a matrix is built. The matrix rows represent the corpus nouns, whereas columns represent the nouns in the dictionary definitions for the corpus nouns. All duplicate values are eliminated and the matrix has values that can be only one and zero. When k-means, c-means and hierarchical clustering methods are implemented, very meaningful clusters have been observed, showing that the semantic relationship between words can be seen through the use of dictionary. Nonetheless, there was a problem such that there were many clusters that included only one word and some clusters had disproportionately many words. Among hierarchical clustering alternatives, *euclidean* and *cosine similarity* metrics are performed and it has been noticed that the cosine similarity metric, to a some degree, decreased the outlier problem with a higher success than euclidean one. This may be attributed to the fact that cosine metric measures the similarity between two nodes (words) in terms of the angle between the lines through which nodes are attached to origin applying also normalization, instead of simply measuring the distance between two nodes on geometrical plane through euclidean distance. Instead of these methods, a simpler statistical algorithm is developed eliminating algorithms that take into account clustering.

3.4. Simple Frequency Matrix and Context Analysis Algorithms using Türk Dil Kurumu (TDK) Dictionary

In Concept Mining field of NLP, one of the most resorted techniques is the one that takes frequency into account. It makes sense since the general idea of a document can be extracted through the words that are frequent in this document. If a word is found only once, or twice such as *attorney* in a lengthy document, this word may not be a top candidate concept amongst many words. So in this thesis frequency measure is used. But taking into account only the frequent words that are present in the document may not be

sufficient. For example there may be words such as *football*, *basketball*, and *handball* in the document being examined. Just thinking of the words in the document as concepts may be wrong, because a concept may be present, or be absent in the document. So in this thesis, taking into account that concepts may not be present in the document, the TDK Dictionary is used.

3.4.1. The Structure of the Dictionary

So far, in the Concept Mining field, although the use of many language models (LM) such as LDA have been seen to be beneficial [15], the use of lexical databases with AI methods such as clustering has been more prevalent [16] and it gives higher success rates. The most widely used lexical source is WordNet, which provides synsets that are composed of many properties. Synsets are a set of relations through which analogies can be made between words. For example, the synset relation *synonymy* implies that two words have the same meaning, such as the relation between *attorney*, and *lawyer*, another relation called *hypernymy* implies that one word has a general meaning for another, such as the relation between *animal*, and *organism*. *meronymy* relation implies that one word is part of the other word such as the relation between *eye*, and *face*, and there are a few more relations.

Among the relations of synsets, the one that is called hypernymy is most widely used for extracting concepts due to that a general meaning of a word can give us a general idea concerned with this word. Some examples of hypernymy relation is shown in Table 3.3.

Table 3.3. Hypernymy examples.

Words	**Hypernyms**
Chihuahua	Dog
Earth	Planet
Animal	Organism
School	Building
Engineering	Profession

So far, in the studies concerned with concept mining, other synset relations besides hypernymy have rarely been preferred and used, due to the fact that is stated above, that is, hypernyms of words can suggest a concept set regarding this document. High levels of hypernymy relations can be used in some algorithms. Taking into account only the one-level hypernymy may not suggest a general concept concerning document, two-level or higher levels may be used. For example two-level hypernymy counterpart of the word *Chihuahua* may be *animal*, since all *Chihuahuas* are dogs, and all dogs are animals. But this study approaches concept mining field in a novel way which has not been done so far: The use of basic language dictionary. This is the case since WordNet has a poor and incomplete structure in Turkish, also the performance of the use of dictionary may excel that of WordNet in some ways.

TDK Dictionary is the official dictionary in Turkey Turkish that is most widely used across the world. In this study, this dictionary is utilized through electronic medium, in XML format. This dictionary, like any others in other languages, is composed of properties as follows:

- Word entries,
- Word categories, such as adjective, noun, etc.,
- Word meanings,
- A usage shown in examples through citation sentences,
- Possible affixes,
- Stress, indicating which syllable must be strongly pronounced,
- Language of origin for the word,
- Compound phrases in which this word entry may be used,
- Proverbs, or idioms making use of this word entry.

Sometimes, some of the properties for a word entry in the dictionary may be absent or may have many values, for example the word *address* may be used in either verb, or noun categories, as for in any language, it is possible that a word may have many grammatical categories. Therefore the specific word category can be defined by the POS tagging, looking into its context. Figure 3.1 shows an example of the word entry *jaguar* and its properties, in XML format of dictionary that we utilized. Some tag elements are

labeled "*undefined*", that is, those tag properties are not defined for this word entry. For example, the tag <*atasozu_deyim_bilesik*> stands for *proverb*, *idiom*, *compound* in Turkish, and the word *jaguar*, as shown in Figure 3.1, is not used in any proverb, idiom or compound, that is why this tag element is defined as "*undefined*".

```
<entry>
    <name> jaguar </name>
    <affix>undefined</affix>
    <lex_class>isim, zooloji </lex_class>
    <stress>undefined</stress>
    <pronunciation> Fransızca jaguar </pronunciation>
    <origin> Fransızca</origin>
  - <meaning>
        <meaning_class>undefined</meaning_class>
        <meaning_text>  Kedigillerden, Orta ve Güney
            Amerika'da yaşayan, postu iri benekli memeli
            türü (Felis onca).</meaning_text>
      - <quotation>
            <author>undefined</author>
            <quotation_text>undefined</quotation_text>
        </quotation>
    </meaning>
    <atasozu_deyim_bilesik>undefined</atasozu_deyim_bilesik>
    <birlesik_sozler>undefined</birlesik_sozler>
</entry>
```

Figure 3.1. A word's properties in the XML format of dictionary from which we benefited.

Among the properties of the dictionary, we overlooked some of them, such as the stress, affixes, origin language, proverb uses, citation sentences and compound phrases, because they may not contribute to the extraction of the concepts from a document. We make use of the words if they are nouns by looking into their word category property, and we benefit from the dictionary definition sentence.

Meaning texts can be used to extract expressive information concerned with the word itself, and can be benefited from for extracting concepts. These meaning texts show the properties of words, as it is the case for WordNet relations as well. These properties may be like hypernymy, meronymy, or synonymy relations between the word entry, and meaning text words. For example, the below dictionary definition for football can be examined:

Football: "A game played by two teams of 11 players each on a rectangular, 100-yard-long field with goal lines and goal posts at either end, the object being to gain possession of the ball and advance it in running or passing plays across the opponent's goal line or kick it through the air between the opponent's goal posts."

For example, the word *game* in the dictionary definition has a hypernymy relation with the word entry *football*. The word *goal* is the aim technique for this game and the word *ball* is the main object that is used in this game, so there are relations between those words as well. The most widely used relations between a word entry and the other words that are in meaning text of this entry can be summarized as follows:

- Synonymy: It is a relation such that two words have equivalent meanings. It is a symmetrical relation. For example, the words *human being* and *person* are synonyms.
- Meronymy: It is a relation such that one of the words is a constituent of another. It is not a symmetrical relation. For example, the words *finger* and *hand* have this relationship.
- Location: It is a relation that shows the location of a word with respect to another. For example, the words *capital* and *country* have this relationship.
- Usability: It is a relation such that one word is used for an aim. For example, toothbrush is used for brushing teeth.
- Effect: It is a relation such that one action (word) leads something to take place. For example, taking medication leads to a healthy state.
- Hypernymy: As mentioned, it is a relation that one word is a general concept of another word. For example the words *dog* and *Golden Retriever* have this relation.
- Hyponymy: It is a relation such that one word has a more specific meaning of another word. For example, the words *teacher* and *profession* have this relationship. It is not to be confused with the meronymy relationship.
- Subevent: It is a relation that one action has a sub-action. For example, waking up in the morning may make one yawn.
- Prerequisite relation: It is a relation that one action is a prerequisite condition for another one. For example, waking up in the morning is a prerequisite condition for hitting the road for job.
- Antonymy: It is a relation such that one word has the opposite meaning of the other word. For example the words expressing emotional states, such as *happy*, and *sad* have this relationship.

The above relations can be used to measure the analogy between words, and as for Turkish it can be clearly seen that using the official dictionary is much more useful since

this is more comprehensive compared with WordNet, synsets of which are constructed poorly and incompletely for this language. Nonetheless, it should be noted that some of the features stated above, such as antonymy, should be discarded when performing concept extraction.

Making analogy between words can be used in algorithms. For example, if one wants to cluster the words and thereafter want to classify the documents, this method would be useful. The analogies between words, which can be seen by the existence of common words in the meaning texts of those words, can make some words be assigned into the same cluster, or into another one. The only possible relation that would be considered harmful when performing clustering that uses the similarity of meaning texts is antonymy. One word may be expressed in the meaning text of its antonym word, due to this common word they would be assigned to the same cluster, which is not sensible. For example, we may look at the below meaning texts of two words.

Table 3.4. Raw dictionary definitions of two words.

Cat: "A small carnivorous mammal *(Felis catus* or *F. domesticus)* domesticated since early times as a catcher of rats and mice and as a pet and existing in several distinctive breeds and varieties."
Lion: "A large carnivorous feline mammal *(Panthera leo)* of Africa and northwest India, having a short tawny coat, a tufted tail, and, in the male, a heavy mane around the neck and shoulders."

The common words in the above two sentences are *carnivorous*, and *mammal*. This may show that these two word entries would be similar in some senses, so they may be assigned to the same cluster. Also other category members can be assigned to the same cluster, for example fruits, such as apple, peach, and cherry. They can be grouped in one cluster, also animals, month names, profession names, electronic devices, and many other specific category members can be grouped in separate clusters.

The relations in the dictionary described above may be algorithmically applied in NLP, but it is important to note that in this study, only nouns are thought of as concepts, so the ones which are not related to the noun category are eliminated. For instance, sub-event

relation takes only verbs into account and makes analogies between these events, hence we overlook this relation.

Since in this work, the official dictionary is utilized, its text has to be parsed and disambiguated. These processes are required since Turkish is an agglutinative language and many inflectional morphemes have to be eliminated. For example the meaning text for the word of *jaguar* in Turkish is as follows:

"Kedigillerden, Orta ve Güney Amerika'da yaşayan, postu iri benekli memeli türü (Felis onca)."

Here, the first word should be returned as *kedigiller*, eliminating the inflectional morpheme *den*, and then this processed word must be utilized in the algorithm. Nevertheless, it is important to note that while parsing operations are performed successfully, words cannot be disambiguated correctly at that high success rate. It is due to that dictionary definitions of words are generally composed of a few words, and due to this data sparseness, averaged perceptron-based algorithm cannot assign very meaningful scores to any possible POS tags.

3.4.2. Context Analysis for Disambiguation

It is possible for word entries in the dictionary to have many different meanings. We have to select the one which is used in the word's context in document. For instance, the word *bank* has many meanings and we have to extract the true meaning text by looking into the document. In order to extract the true meaning, we can perform context analysis [18]. Context analysis in NLP is that we create windows surrounding a word and perform analysis consistent with the words in these windows. The size of windows can vary, but the most widely used ones are generally of 30, 50 and 70 word sizes. These can be called n-grams, for example, if a 30-gram window is to be used, 15 words on the left of the test word, and other 15 words on the right of the test word are taken into account. In this work, 30-grams are used.

The contexts are the words surrounding a test word in corpora. All of these context words are compared with the meaning text words, and if the number of common words is

high compared with that of other meaning texts, this meaning would be chosen as the true meaning. It can be formulized as follows:

$$\text{argmax}_{\text{m}} Similarity(m, c_w) = \frac{CommonCount(m, c_w)}{length(m)} \quad (3.2)$$

The above formula finds the highest similarity score among candidate meaning texts. *w* stands for the corpus noun, m stands for the meaning text, c_w stands for the context of the word *w*, *CommonCount* counts the number of common words that are found in both context and meaning text of a word. Lastly, we have to normalize the score by dividing the score by the number of nouns in the meaning text. It is sensible since a meaning text may contain many words and if the number of common words found are not too high, then the other meaning text with a fewer words should be scored higher and be favoured. The words that are taken into consideration are only nouns.

This context analysis is useful especially when taking into account that many words have more than one meaning. But if the documents' sizes are small, this may be a drawback that is called data sparseness. For example, if a document contains fewer than 30 words, say ten, then the algorithm may fail in performance results. If the context size is increased, more meaningful results can be achieved, but it has the drawback that performance success (time and space complexity) may be lessened. Also it should be noted that the context words do not have to be nouns. They can be adjectives, adverbs, verbs or pertain to other word categories. This is a fair approach, because if we eliminate the non-noun words before making context analysis, then the nouns that are, in fact, far from one another, can be thought of as that they are close to each other, and this would be problematic. When creating n-gram contexts, words of any category, such as adjective, verb, etc. are first taken into consideration, but thereafter when only nouns are selected as probable concepts, words being not nouns are eliminated.

In this thesis there have been developed a few different algorithms and it is seen success rates for different corpora vary in accordance with those algorithms. When looking at overall results, the second algorithm (Section 3.4.4) excels the performance of the first algorithm (Section 3.4.3) performed in most of the corpora.

3.4.3. Simple Frequency Algorithm (Alternative 1)

Concepts of a document are generally related to the words that abound in that document, therefore we had to take the frequency factor into consideration. For instance, if we encounter a document that abounds with the word *football*, we may be inclined to think that the concept of this document would be concerned with the topic of *sport*. Taking frequency into account, accordingly a statistical method was developed, that extracts concepts favouring the words that appear more frequently in the document.

We first take all the nouns in the document(s) and label them as pre-concepts. Here we eliminate other types of words, such as adjectives and verbs. Then we can start building a matrix. This matrix has rows representing the nouns encountered in the document and columns representing the nouns encountered in the meaning text sentences of those row words. But we also have to take into account that the row words should be added as column items once. For example, the word *football* may be quite frequently found in the document, therefore it should be regarded as a probable concept as well.

The cells in the matrix are filled as follows: After we built the matrix, we fill the cells by one or zero values, depending on whether the column word appears in the row word's dictionary definition or not. Then we perform the frequency operation: We multiply all the cell values in the matrix by the corresponding row word frequency. For instance, if the word *football* is encountered 10 times in a document and its meaning text nouns in the dictionary are *sport*, and *team*, then those columns' (*sport*, *team* and *football*) values in the correspondent row *football* would be updated as 10, whereas the other columns would be updated as zero. Then the cell values in the matrix are multiplied by the row word's scope and first location properties. The term scope purports how a word is distributed over a document. If a word is encountered in only a paragraph in the document, its scope would be assumed to be small. On the contrary, if a word is encountered in the different sections of the document, say first and last paragraph, its scope would be assumed to be large. First location indicates the first location of a word, if it is encountered in initial positions of the document, its value is higher, otherwise it is lower. A logarithmic approach is utilized when performing these two functions.

Finally, the column values are summed up in the matrix and we think of the column word that produces the highest summation as probable concept. This is meaningful since the concept may or may not occur explicitly in the document, therefore the use of dictionary would be beneficial. An example showing the mapping of terms into concepts is given in Figure 3.2.

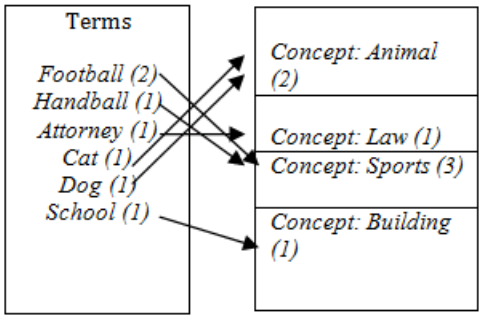

Figure 3.2. An example showing the mapping of document words into concepts.

In Figure 3.2, the column on the left is a representative of words encountered in the document, whereas the column on the right includes the nouns encountered in their meaning texts. Since the words *football* and *handball* are frequent in the above example and the word *sports* occurs explicitly in their meaning texts, its score would be three and this word would be assigned as the probable concept for the document.

As mentioned, we benefit from dictionary to extract concepts from documents, but instead of using just the nouns found in the dictionary definitions, also a hierarchical data structure that contains two, three and four levels is built. Consistent with this structure, the main word entry is atop the hierarchy, then the meaning text nouns of this word is in the lower level, whereas the respective meaning text nouns of these dictionary definition nouns are in the lower levels. An example of this data structure with three-levels is depicted in Figure 3.3.

This hierarchical structure may have some specific features, for example each word in different levels may be assigned a different coefficient and we may take this coefficient factor into account when building up the matrix. If we construct three-level hierarchies through the dictionary, we may assign high values for the top level words and low values

for lower level ones. This is the case because the semantic relationship between the main word and the lower level nouns weakens while going down through the hierarchy structure.

We multiplied the top-level words in the matrix by 1, the second-level words by 0.5 and the lowest-level words by 0.25. We utilized this geometric approach, since the meaning text nouns' frequencies increase geometrically from one level to the below one. But we noticed that three-level structure gives slightly higher results compared to that of two-level structure, so we preferred three-level structure with coefficients producing higher precision. Also four-level structure gave worse results than three-level one, so using a three-level structure was considered the best alternative.

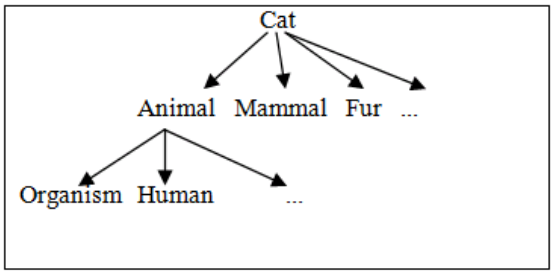

Figure 3.3. A hierarchical data structure with three-levels of the word *cat* in the dictionary.

The matrix cells are filled, as mentioned above, without taking frequency into account and the results yielded were much less successful. That shows the importance of taking frequency into account. The pseudo-code of this algorithm is given in Figure 3.4.

We also have to take into account that some words are quite common in the dictionary, such as *situation*, *thing*, *person* and so on. Here the top 1% most frequent words found in the dictionary definitions are chosen as stop-words and are eliminated. Generally tf-idf is used for elimination of words, but since we make use of the dictionary as a base, top words elimination is seen to be sufficient.

Algorithm: Extracting concepts through simple frequency using dictionary

Input

F1: Documents in corpus

Output

F2: Concepts of documents

Begin

1: *L* <- Assign *F1* to the list
2: **for** each document *i* in *L*
3: *Matrix* <- Ø
4: **for** each word *j* in document *i*
5: *Meaning* <- Meaning text nouns of word *j*
6: Add word *j* to *Meaning*
7: **for** each word *k* in *Meaning*
8: $Matrix(j, k) = Freq(j) \times FirstLoc(j) \times Scope(j)$
9: **end for**
10: **end for**
11: Fill the cells in *Matrix* by zero value which have no value assigned
12: *Matrix* <- Remove Duplicate Row and Columns of *Matrix*
13: *List* <- sum(*Matrix* columns)
14: *List* <- sort(*List*)
15: Add column words, corresponding to top(*List*), to *F3*
16: **end for**

End

Figure 3.4. Pseudo-code for the extraction of concepts using dictionary that takes into account frequency factor.

3.4.4. Frequency and Context Algorithm (Alternative 2)

Although it is noticed that the algorithm 1 developed for this thesis stated above produced meaningful concepts, drawbacks can be clearly seen. For example, we may assume there is a document containing the noun *football* and there is no other noun, with its meaning text in the dictionary being as follows:

"A game played by two teams of 11 players each on a rectangular, 100-yard-long field with goal lines and goal posts at either end, the object being to gain possession of the ball and advance it in running or passing plays across the opponent's goal line or kick it through the air between the opponent's goal posts."

According to the algorithm stated in Section 3.4.1, we take the nouns in this meaning text into consideration and build up a matrix containing those nouns, including the word *football*. Since the word *football* is seen three times, the column labeled *goal* has a value of three as well and at the end the probable concept may be the word *goal*. Other concepts may be *game*, *team*, *line* and other nouns that are encountered in the meaning text. (This is the case since the matrix would be of size 1 x *CountNoun(MeaningTextOf(Football)*), indicating that there is only one noun, that is *football* in the document.) Having a concept that is *goal* for this document would not be quite reasonable (also we can assume that properties of first location, and scope are ignored in this example), therefore the algorithm is modified in the following manner:

All the dictionary meaning text nouns would not be useful in determining the general idea stated by the main word, so some of those nouns have to be eliminated. In order to determine which meaning text noun is relevant to the main word in its context, a corpus-based context analysis is used. There are a few corpora and for each corpus, a 30-word window size context analysis is performed, that is, 15 words on the left of the test words and 15 words on the right of the test words are looked up. Hereby, the context words which are not nouns are eliminated, because we think of concepts as only nouns. Then it is assumed that if a context word explicitly occurs in the meaning text of the main word in dictionary, we take this context word into consideration. After scanning the whole corpus, whichever context word is most frequently found, given that context word is also seen in

the meaning text of the main word, this word is added as a column word in the matrix corresponding to the row word. Then, similar to what we have done in Section 3.4.1, we multiply the row elements values by the frequency, first location, and scope properties of the row representative word and sum up the columns values. Whichever column value has the maximum value, we define that column representative word as the probable concept. In this case, we take mostly two words for each word in the document: The word itself and the word in its dictionary definition that is most widely encountereed in its contexts in the corpus. The stop words being present in the TDK Dictionary are eliminated.

This approach is sensible, since all meaning text nouns may not be useful in determining the general idea, that is concept, of a word. Also the corpus-based approach shows that the most relevant word in the meaning text of a test word is extracted through the context analysis. Selecting at most two words, that are the word itself and the most frequently occurring word in its contexts that also appears in the meaning text of the row word in the matrix increased the success rate to a great extent for three corpora, rather than taking into account all nouns in the word's dictionary definition. Pseudo-code of this algorithm is given in Figure 3.5.

3.5. Simple Illustrations of the Methodology

Simple frequency algorithm takes into account the nouns in the document, their meaning text nouns explicitly occurring in the dictionary, and their frequencies. For example we may assume there are two nouns in the document, that are *tiger* (which stands for *kaplan* in English), and *monkey* (which means *maymun* in Turkish). The dictionary definitions of these words are shown in Table 3.5.

Table 3.5. Dictionary definitions for two words, that are *kaplan* and *monkey*.

Kaplan: Kedigillerden, enine siyah çizgili, koyu sarı postu olan, Asya'da yaşayan çevik ve yırtıcı hayvan (Felis tigris).
Maymun: Dört ayaklı, iki ayağı üzerinde de yürüyebilen, ormanda toplu olarak yaşayan, kuyruklu hayvan.

Algorithm: Extracting concepts through simple frequency using dictionary

Input

F1: Documents in corpus

Output

F2: Concepts of documents

Begin

1: *L* <- Assign *F1* to the list
2: **for** each document *i* in *L*
3: *Matrix* <- Ø
4: **for** each word *j* in document *i*
5: Add the word that is in the dictionary definition of *j* and is the most frequently encountered noun in the corpus into *Meaning*
6: Append word *j* to *Meaning*
7: **for** each word *k* in *Meaning*
8: $Matrix(j, k) = Freq(j) * FirstLoc(j) * Scope(j)$
9: **end for**
10: **end for**
11: Fill the cells in *Matrix* by value zero which have no value assigned
12: *Matrix* <- Remove Duplicate Row and Columns of *Matrix*
13: *List* <- sum(*Matrix* columns)
14: *List* <- sort(*List*)
15: Add column words, corresponding to top(*List*), to *F2*
16: **end for**

End

Figure 3.5. Pseudo-code for the extraction of concepts using dictionary that takes into both frequency factor and context analysis.

Then we are to build the matrix, row words of which are the document words whereas the column words are the nouns found in their meaning texts. Table 3.6 shows this

matrix, it should be taken into account that duplicate nouns are removed. (In this example the first location, and scope properties are ignored to make it be more comprehensible and less complex.)

Table 3.6. Matrix constructed with the words *tiger* and *dog* in accordance with simple frequency algorithm.

	Kedigiller	çizgi	post	hayvan	orman	kuyruk	kaplan	Maymun
Tiger	1	1	1	1	0	0	1	0
Monkey	0	0	0	1	1	1	0	1
Summation	1	1	1	2	1	1	1	1

Among the column words in Table 3.6, the noun *hayvan* has the highest value, that is two, and this is labeled as the top concept. The words *kaplan* and *maymun* are also added as column words once since document words can be probable concepts. But it should be taken into consideration that some words such as *post* and *çizgi* may not play a role in determining general concept of this document, so they are eliminated. Second alternative, that is frequency and context algorithm may produce better results counting this factor compared with the first alternative.

Second algorithm simply takes into account the document words and one meaning text noun for each document word that is most commonly found in the contexts of those document words in the whole corpus. A simple example can be examined in Table 3.7.

Table 3.7. Matrix constructed with the words *tiger* and *dog* in accordance with the frequency and context algorithm.

	hayvan	kaplan	maymun
Tiger	1	1	0
Monkey	1	0	1
Summation	2	1	1

Consistent with the matrix shown above, the most frequent word in the contexts of both words that are *kaplan*, and *maymun* is *hayvan*. Other words are eliminated since there

would be only one word that is most frequently found in the corpus. Also the words *kaplan* and *maymun* are added as column words since they appear in document. Here again the highest score is that of word *hayvan*, so the concept of this document may be labeled *hayvan*.

Table 3.8. Content of a document from Forensic Decisions corpus.

T.C.
YARGITAY
6. Ceza Dairesi

YARGITAY İLAMI

Esas No: 2001/10772
Karar No: 2001/14183
Tebliğname : 6/12620

ÖZET: Sanığın, staj yaptığı bankanın müşterisinin banka kartıyla şifresini ele geçirip ATM'den para çekmekten ibaret eylemi TCY.nın 525/b-2.maddesine uyan suçu oluşturur

Dolandırıcılıktan sanık H.G ve M.Ö'nin yapılan yargılanmaları sonunda: Mahkumiyetlerine ilişkin İSTANBUL 6.Ağır Ceza Mahkemesinden verilen 22.11.1999 tarihli hükmün Yargıtay'ca incelenmesi sanık Hasan müdafii ile duruşmalı olarak sanık Mehmet müdafii tarafından istenilmiş olduğundan dava evrakı C.Başsavcılığından onama isteyen 15.6.2001 tarihli tebliğname ile 28.6.2001 tarihinde daireye gönderilmekle tayin edilen günde yapılan duruşma sonunda okunarak gereği görüşülüp düşünüldü.

Sanık H.G. müdafiinin yasal süreden sonraki temyiz isteminin CMUK.nun 317.maddesine göre REDDİNE Sanık M.Ö'e ilişkin temyiz incelemesine gelince Adı geçenin, staj yaptığı bankanın müşterisi K.A nın banka kartıyla şifresini ele geçirip daha sonra ATM.den para çekmekten ibaret bulunması karşısında, eyleminin TCK.nun 525/b-2.maddesine uyan suçu oluşturacağı gözetilmeden,unsurları bulunmayan dolandırıcılıktan mahkumiyetine karar verilmesi Bozmayı gerekçe olarak BOZULMASINA ilişkin oybirliğiyle alınan karar 21.11.2001 günü Yargıtay C.Savcısı önünde, sanık müdafiinin yokluğunda açıkça ve yöntemince okunup anlatıldı.

Table 3.8 shows content of a document from Forensic Decisions corpus, whereas Table 3.9 shows the concepts that are extracted algorithmically from this document, in a

decreasing order of importance, as an example. It can be seen most of the concepts extracted are meaningful. Some words that do not appear in the document can also be probable concept candidates. For example the word *hüküm* does not appear in the document shown in Table 3.8, but algorithm defines this word as a concept as shown in Table 3.9.

Table 3.9. Top 15 Concepts extracted algorithmically from the document shown in Table 3.8.

1.	sanık
2.	suç
3.	banka
4.	faiz
5.	usul
6.	daire
7.	ceza
8.	staj
9.	hizmet
10.	müşteri
11.	hüküm
12.	telefon
13.	kart
14.	şifre
15.	eylem

4. EXPERIMENTS AND EVALUATION

4.1. Corpora

In this study, four corpora are processed and algorithm developed in accordance with this work extracts expressive concepts from these corpora. These four corpora are mentioned in detail in Chapter 3. These corpora are processed through two algorithms, with the former one taking into account dictionary structure and properties of words, whereas the latter one taking into account also context analysis. On average, the second algorithm (alternative 2) is seen to produce better performance results.

4.2. Evaluation Metrics

As evaluation of results, there have been developed many metrics in the domain of science, engineering and statistics. The most widely used ones are precision, recall and accuracy. In the domain of NLP, they are the commonly used evaluation metrics as well. These metrics can be formulized as follows:

$$Precision = \frac{number\ of\ true\ positives}{numer\ of\ true\ positives + false\ positives} \quad (4.\ 1)$$

$$Recall = \frac{number\ of\ true\ positives}{number\ of\ true\ positives + false\ negatives} \quad (4.\ 2)$$

$$Accuracy = \frac{number\ of\ true\ positives + number\ of\ true\ negatives}{number\ of\ true\ positives + false\ positives + false\ negatives + true\ negatives} \quad (4.\ 3)$$

Precision (also known as positive predictive value) simply is the fraction of retrieved instances which are relevant, while recall (also called sensitivity) is the fraction of relevant instances that are retrieved. For example, a search engine takes queries and if there are 10 documents which are relevant for a query and amongst these documents three of them are returned, recall is 3 / 10. If search engine shows five documents as the top results, then precision value would be 3 / 5. In this thesis study, precision and accuracy are utilized as

evaluation metrics, since in NLP domain, as mentioned, those are the most widely used and expressive metrics. Table 4.1 examines those metrics as shown below.

Table 4.1. Evaluation metrics.

		Condition as determined by *Gold* standard		
		True	False	
Test Outcome	Positive	True positive	False positive	Positive predictive value or Precision
	Negative	False negative	True negative	Negative predictive value
		Recall or Sensitivity	Specificity	Accuracy

4.3. Evaluation Method using Comparison Windows

In this thesis work, files were created which contain concepts for each file in corpora that are extracted through the algorithm developed. These files include the top 15 concepts that are produced by the algorithm. The concept terms that have a higher score assigned in accordance with the matrix algorithm than that of others are labeled as 'top concepts'. In order to evaluate the precision of those assigned concepts, totally 368 files in four corpora are examined and concepts are manually extracted. Then the concepts that are extracted manually and algorithmically are compared with one another. In manually extraction manner, all the files in the corpora are read by two humans, and hereafter concepts of those files are lined up in a decreasing order of importance. This comparison is performed with windows, sizes of which are chosen as three, five, seven, eight, nine, ten, and fifteen words. For different corpora, the window comparison sizes used are as follows:

- Forensic Decisions Corpus: Three, five, seven, ten and fifteen window comparison sizes are used.
- Forensic News Corpus: Three, five, seven and eight window comparison sizes are used.

- Sports News Corpus: Three, five and seven window comparison sizes are used. It has many different topics concerned with sports topics.
- Gazi Corpus: Three, five, seven and nine window comparison sizes are used.

Also for all corpora, the top concepts found algorithmically were compared with all the concepts extracted manually, which is called unlimited comparison.

To illustrate this comparison, Table 4.2 can be examined.

Table 4.2. An example showing the top three concepts in two documents.

Documents	**Algorithm**	**Manual**
Document 1	Sport, Game, Match	Sport, Match, Politics
Document 2	Court, Attorney, Judge	Attorney, Accused, Match

Table 4.2 shows the top three concepts for two documents, extracted both manually and algorithmically. In the first document, it can be seen that the success rate (precision) is 2 / (2 + 1) = 0.66, since there are two words in common, which are *sport* and *match* that are found both in concept clusters extracted manually and algorithmically. However, the word *game* is not in the top three concept set extracted manually, so it decreases the success rate. In Document 2, the success rate is 0.33, since only the word *attorney* is common among the three top concepts. This is an example taking into account comparison window size which is three, also other comparisons can be similarly made taking into consideration different sizes.

The evaluation result precisions vary from one corpus to another one, showing that the concepts extracted can be corpus-biased. It is seen that higher precision results are achieved for *Forensic Decisions,* and *Forensic News* corpora, whereas the precision results for corpora *Sport News*, and *Gazi* are evidently lower. This may be due to that topic distribution in the former two corpora is not as diverse as that in the latter two corpora. The topics in the corpus *Sport News* are mainly concerning *football* albeit there are many other

topics about sports. As for the corpus *Gazi* there is not such a specific topic. The topics in this corpus are very diverse, some of them including reports concerned with different engineering fields or architecture. Since the second algorithm is corpus-based, having no common topics leads the precision to decrease.

The precision results, taking account of unlimited comparisons for different corpora are depicted in Figures 4.1-4.5:

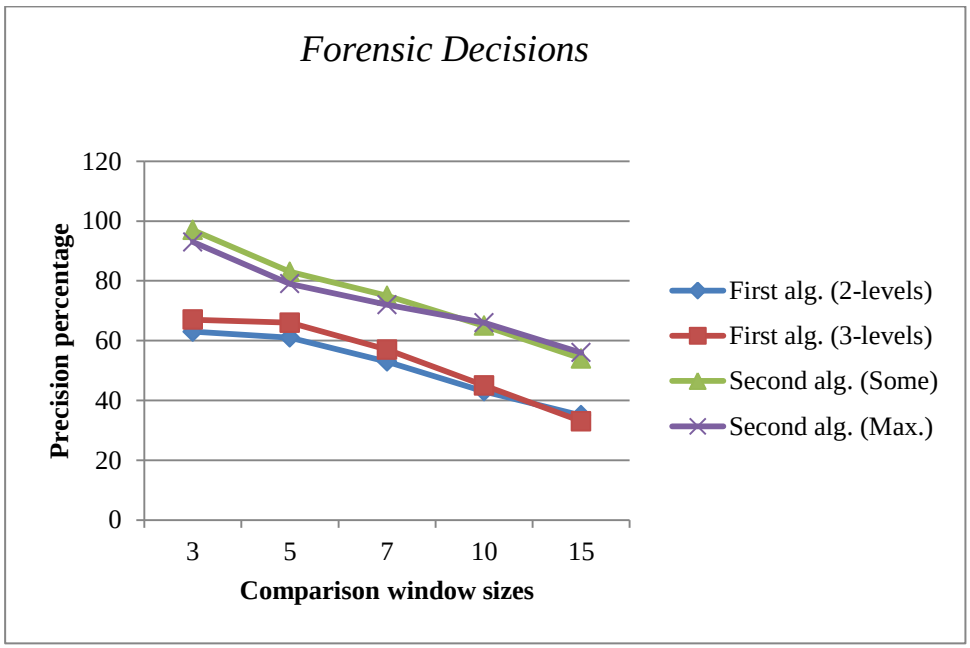

Figure 4.1. Precision percentages for Forensic Decisions corpus in accordance with unlimited comparison window sizes.

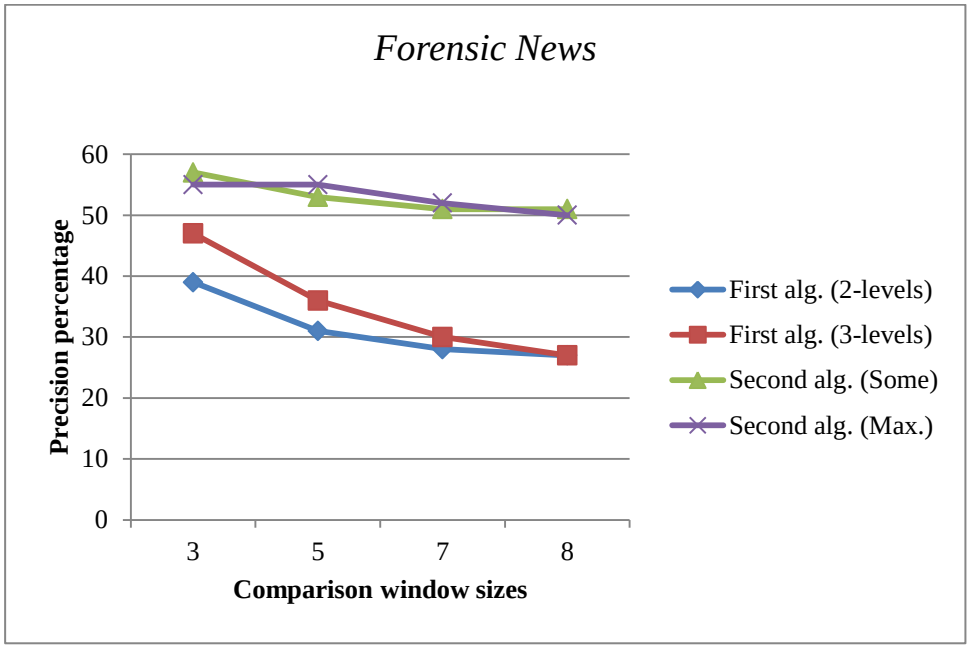

Figure 4.2. Precision percentages for Forensic News corpus in accordance with limited comparison window sizes.

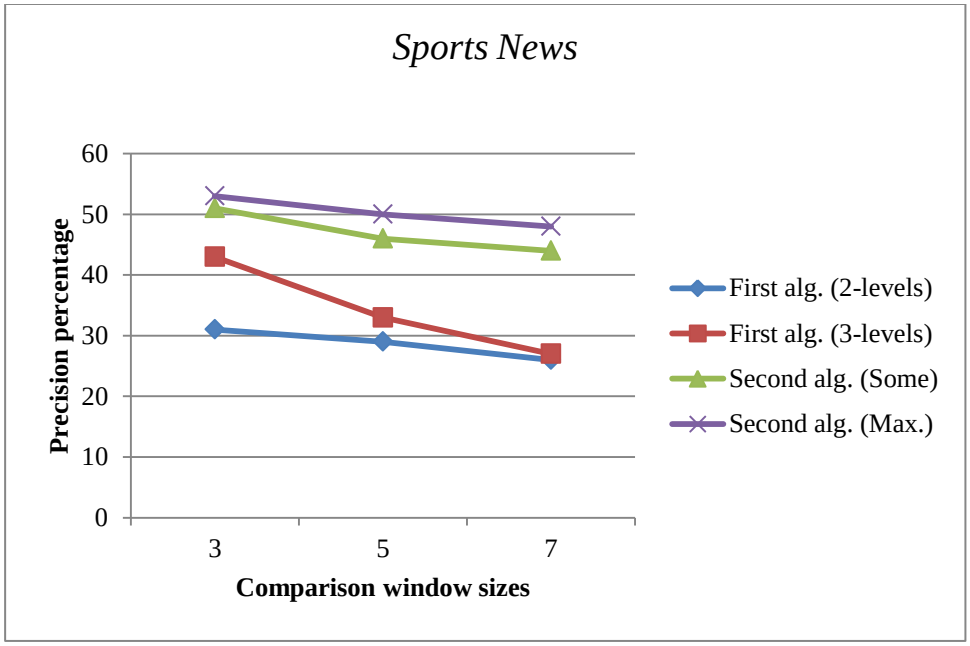

Figure 4.3. Precision percentages for Sports News corpus in accordance with unlimited comparison window sizes.

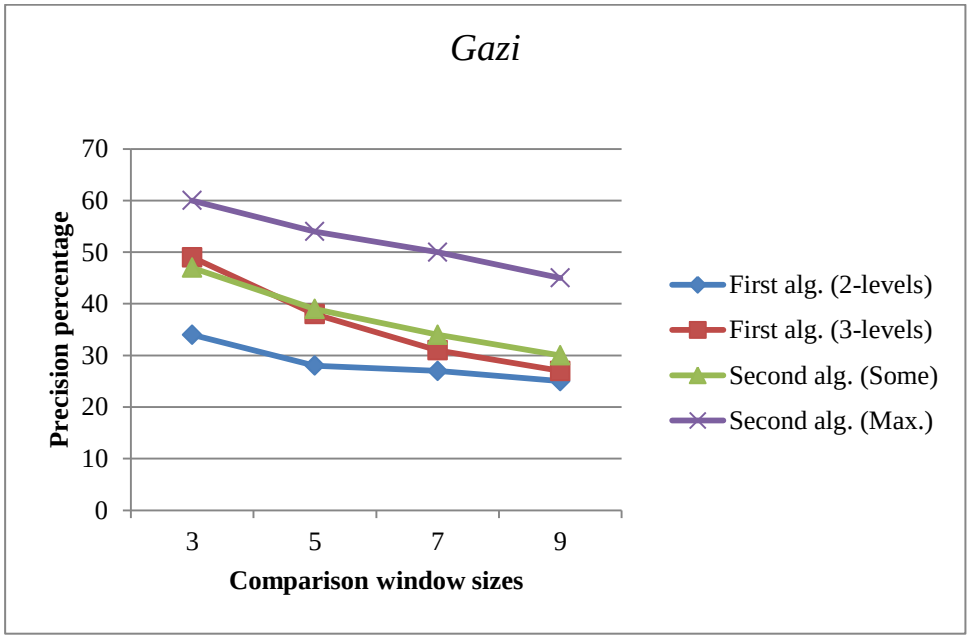

Figure 4.4. Precision percentages for Gazi corpus in accordance with unlimited comparison window sizes.

In above figures, First alg. (two-levels) stands for the first alternative algorithm, according to which a matrix is built as explained in Section 3.4.3, with a hierarchical structure with two-levels being taken into consideration. First alg. (three-levels) takes account of three-level hierarchical structure built through dictionary, with different coefficients for different levels, and frequency. Second alg. (Max.) is the second approach developed, according to which for each document noun, the noun itself and another noun that is both most widely found in the contexts of the document word in corpora, and that explicitly occurs in the meaning text are taken into account, that is, at most two words for each document noun are used for each row in the matrix. Second alg. (Some) takes into account, for document words, all the nouns appearing in both the meaning text of document nouns, and contexts. Some meaning text nouns are eliminated due to that they are not present in the contexts of the document nouns in corpora. Also some words are assigned higher scores in that they are more widely found in the contexts of document words in corpus. As can be clearly noticed, the first algorithm alternatives produces unsuccessful results, because, as mentioned, all meaning text nouns may not represent the general meaning, that is concept, of a word. Therefore some elimination might lead to amelioration occurring in results.

For different corpora, algorithms produce different precisions. The highest precision results are achieved through the second algorithm (max.) for the corpora Gazi, Sport News, and the second algorithm with two alternatives for the corpora Forensic Decisions and Forensic News. When taking account of the highest precisions obtained using unlimited comparison window size, precision, on average, is 52.1% for the first algorithm (one of the sub-algorithms is selected whichever gives the highest accuracy results), and 63.97% for the second algorithm.

An example of comparing the precision results for different corpora, selecting the window size as seven, unlimited, is shown in Figure 4.5. That is, top three concepts found algorithmically are compared with all concepts extracted manually. An important point here to note is that Forensic Decision success results excel the other ones to an extreme degree, whereas Gazi corpus success rates are relatively low. It may be due to that, as stated before, a single topic is encountered in all the documents of the Forensic Decisions corpus, whereas Gazi corpus includes many different topics distributed over its documents, such as engineering, scientific, or architectural reports and tutorials.

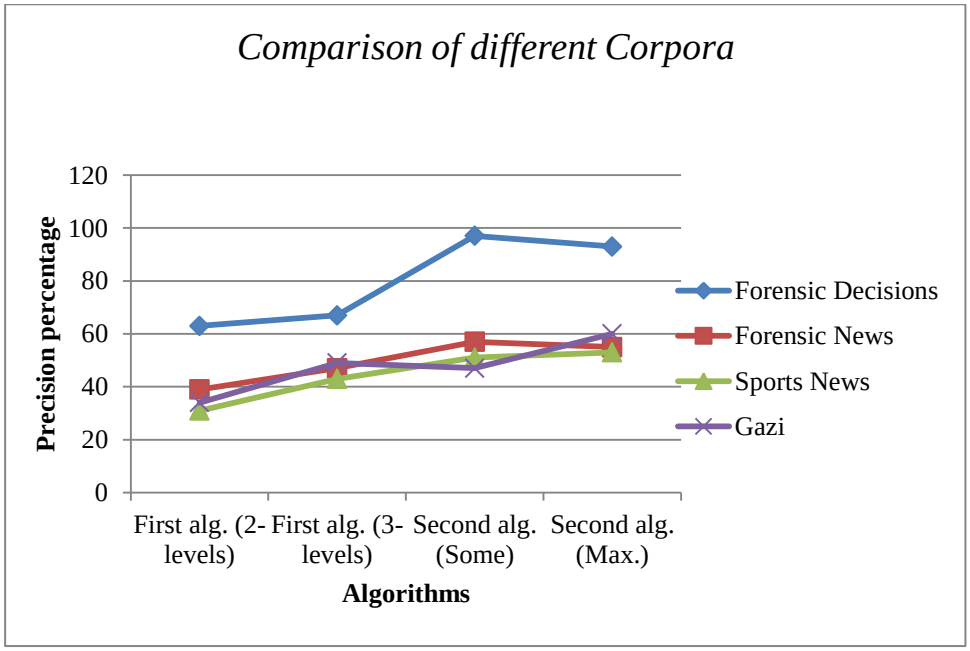

Figure 4.5. Comparison of different corpora in accordance with different algorithms, taking into account three vs. unlimited approach.

5. CONCLUSION

Concept mining is a field of NLP that can be used in medical applications, forensic cases, financial systems such as that of banks, text categorization, search engine algorithms and many other domains. Its importance is increasing since the size of data and documents in electronic medium is growing to an extreme degree, and conceptual information from those electronic materials that may be in textual, visual and audio form, need to be extracted through computerized, automatic methods in an efficient way. Most of the data from which concepts are extracted are in textual form in this domain, whereas concept extraction from visual and audio materials are rarely used compared with the former one. In this thesis, only textual materials are processed for concept extraction.

Many algorithms have been used for extracting concepts so far, but the most commonly used ones are statistical and NLP methods. These methods include SVM, HMM, LSA, clustering and many more algorithms which can ease the extraction operation of the concepts. Possibility of human intervention in NLP makes it more beneficial and useful than making use of statistical methods.

The majority of algorithms used in concept mining domain makes use of AI methods. During this thesis study, machine learning methods such as clustering is implemented, but seeing that no meaningful and successful results could be achieved, these methods are dismissed. Instead, a simple, novel statistical method utilizing the official Turkish dictionary is developed. Two methods have been developed for this thesis work. The former one takes into account all the words in meaning text of a word that explicitly occurs in the document when trying to extract concepts from a document, while the latter one takes account of only the nouns in the document itself, and an extra word that appears in the meaning texts of each of those nouns, which is most commonly found in the their contexts in the corpus. The latter one makes use of the second approach, that is context analysis whereas the former one does not follow such an approach.

In accordance with this algorithm, also some features of the words are taken into account besides using dictionary. These features include the frequency, first location, and scope factors of the terms. This is a reasonable approach, since the general idea, that is concept of a document is generally related to the words that are most widely found in this material. Also other location properties of words may carry a lot of weight with the general idea, that is concept of these documents.

The two algorithms developed for this thesis produces successful results, nevertheless on average, the second alternative gave higher precisions for four corpora. The first algorithm gave a precision result of 52.1%, whereas the second one produced a precision rate of 63.97%. This is the case since the first algorithm takes account of all the dictionary definition nouns of the document nouns. All of these definition nouns may not contribute to the extraction of expressive, general ideas from documents.

Many studies are carried out concerning concept mining for the most widely spoken languages, such as English and Spanish, but as for Turkish, it has still been an immature topic and there have been only a few studies performed in this area. Taking into consideration that the results achieved in this thesis study are high, it may be used for extracting concepts from Turkish documents in other corpora.

As a future work, this study may be improved by taking account of other factors. For example, verbs can also be taken into consideration, because verbs can give a general idea concerned with the document. Verbs are considered to have a core importance in the sentence structure such that all other words semantically depend on them. Therefore, thinking of them as probable concepts may be beneficial. Also noun phrases can be benefited from in this regard. Nevertheless, since through the parser and disambiguator tools the noun phrases cannot be extracted, this approach had to be dismissed in this study. Making use of grammatical cases, such as subject, and object cases can contribute to extracting more meaningful concepts, but since there are not such a Turkish grammatical case identifier tool or program we know, we had to dismiss this approach as well.

Another future work would be done requiring that initial algorithms (Section 3.3) be changed and be improved. K-means, c-means and hierarchical clustering methods

produced unsuccessful results for sentence and window co-occurrence algorithms, hence those methods are dismissed. But for the algorithm that utilizes dictionary, clustering methods can be enhanced. In this thesis study, a corpus-based approach is used, but a new algorithm can be developed that approaches the whole TDK Turkish dictionary as a training data. Consistent with this new algorithm, dictionary word entries can be semantically related to one another through the common words in their dictionary definitions. Since dictionary size is quite bigger than that of corpora, this would constitute a better training data such that after clustering, any word in documents can be assigned to a cluster, since the official dictionary anyway includes all the words in the corpora. Clusters finally can also be homogeneous and their word densities may not differ much from one another.

REFERENCES

1. E., Zalta, "Fregean Senses, Modes of Presentation, and Concepts", *Philosophical Perspectives*, Vol. 15, pp. 335-359, 2001.

2. SPSS Inc., "Mastering New Challenges in Text Analytics", *SPSS Technical Report*, MCTWP-0109, 2009.

3. F., Kalaycılar and I., Cicekli, "TurKeyX: Turkish Keyphrase Extractor", *ISCIS '08. 23rd International Symposium*, 27-29 October, 2008.

4. N., Pala and I., Cicekli, "Turkish Keyphrase Extraction Using KEA", *Proceedings of 22nd International Symposium on Computer and Information Sciences (ISCIS 2007),* Ankara, Turkey, 2007.

5. V., Faber, J. G., Hochberg, P. M., Kelly, T. R., Thomas and J. M., White, "Concept Extraction – A Data-Mining Technique", *Los Alamos Science*, 1994.

6. N. A., Bennett, Q., He, C. T. K., Chang and B. R., Schats, "Concept Extraction in the Interspace Prototype", Technical Report, Dept. of Computer Science, University of Illinois at Urbana-Champaign, Champaign, IL, 1999.

7. M. F., Moens and R., Angheluta, "Concept Extraction from Legal Cases: The Use of a Statistic of Coincidence", *International Conference on Artificial Intelligence and Law*, ICAIL, ACM, 2003.

8. M., Uzun, *Developing a Concept Extraction System for Turkish*, M.S. Thesis, Boğaziçi University, 2011.

9. Z., Elberrichi, A., Rahmoun and M. A., Bentaalah, "Using WordNet for Text Categorization", *The International Arab Journal of Information Technology,* Vol. 5, No. 1, 2008.

10. H., Liu and P., Singh, "ConceptNet - A Practical Commonsense Reasoning Tool-Kit", *BT Technology Journal,* Vol. 22, No. 4, 2004.

11. P. M., Ramirez and C. A., Mattmann, "ACE: Improving Search Engines via Automatic Concept Extraction", *Information Reuse and Integration*, 2004.

12. Z., Chengzhi and W., Dan, "Concept Extraction and Clustering for Topic Digital Library Construction", *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2008.

13. H., Sak, T., Güngör and M., Saraçlar, "Turkish Language Resources: Morphological Parser, Morphological Disambiguator and Web Corpus", *GoTAL 2008*, vol. LNCS 5221, pp. 417-427, Springer, 2008.

14. H., Sak, T., Güngör and M., Saraçlar, "Morphological Disambiguation of Turkish Text with Perceptron Algorithm", *CICLing 2007*, vol. LNCS 4394, pp. 107-118, 2007.

15. L., AlSumait, D., Barbar´a and C., Domeniconi, "On-Line LDA: Adaptive Topic Models for Mining Text Streams with Applications to Topic Detection and Tracking", *ICDM '08 Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, 2008.

16. D., Pennock, K., Dave and S., Lawrence, "Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews", *Proceedings of the Twelfth International World Wide Web Conference (WWW'2003),* ACM, 2003.

17. E., Alpaydın, *Introduction to Machine Learning, 2e,* The MIT Press, London, England, 2010.

18. K., Çelik and T., Güngör, *A Comprehensive Analysis of using Semantic Information in Text Categorization*, M.S. Thesis, Boğaziçi University, 2009.