DEVELOPING A CONCEPT EXTRACTION SYSTEM FOR TURKISH

by

Meryem Uzun-Per

BS, Computer Engineering, Istanbul Technical University, 2007

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

2011

ACKNOWLEDGEMENTS

To my mother and father.

I would like to express my sincere gratitude to Assoc. Prof. Tunga Güngör for his invaluable guidance and help during the preparation of this thesis. I would like to express special thanks to Dr. Hidayet Takçı for giving me inspiration and guiding me during my studies. I am also grateful to Assoc. Prof. İlyas Çiçekli for sharing the *Gazi* corpus with me.

I am grateful to TÜBİTAK-BİDEB for awarding me with their graduate scholarship 2210, which helped me to concentrate better to my graduate education. I also gratefully acknowledge the financial support of Bogaziçi University Scientific Research Fund (BAP) under the grant number 5187 and TÜBİTAK under the grant number 110E162.

I also thank Dean of Computer and Informatics Faculty Prof. Eşref Adalı, and Prof. Fikret Gürgen for kindly accepting being in my thesis committee.

I am grateful to my mother and father for supporting my education till this age. Their endless love, patience, understanding, compassion and self-sacrifice cannot be expressed by words. My siblings, especially Elif, also have a contribution in this process. I love them very much.

Finally, I would like to present my thanks to my husband who attended to my life in the last year of my graduate study. I concentrated on my thesis better by his love and support.

ABSTRACT

DEVELOPING A CONCEPT EXTRACTION SYSTEM FOR TURKISH

In recent years, due to growing vast amount of available electronic media and data, the necessity of analyzing electronic documents automatically is increased. In order to assess if a document contains valuable information or not, concepts, key phrases or main idea of the document have to be known. There are some studies on extracting key phrases or main ideas of documents for Turkish. However, to the best of our knowledge, there is no concept extraction system for Turkish although there are some studies for foreign languages.

In this thesis, a concept extraction system is proposed for Turkish. Since Turkish characters do not fit with the computer language and Turkish is an agglutinative and complex language a pre-processing step is needed. After pre-processing step, only nouns of corpus, which are cleared from their inflectional morphemes, are used because most concepts are defined by nouns or noun phrases. In order to define documents with concepts, clustering nouns is considered to be useful. By applying some statistical methods and NLP methods, documents are identified by concepts. Several tests are done on the corpus that is tested in the bases of words, clusters, and concepts. As a result, the system generates concepts with 51 per cent success, but unfortunately it generates more concepts than it should be. Since concepts are abstract entities, in other words they do not have to be written in the texts as they appear, assigning concepts is a very difficult issue. Moreover, if we take into account the complexity of the Turkish language this result can be seen as quite satisfactory.

ÖZET

TÜRKÇE İÇİN KAVRAM ÇIKARMA SİSTEMİ GELİŞTİRİLMESİ

Erişilebilir elektronik verinin ve ortamın son zamanlarda hızla artmasıyla, elektronik dokümanları otomatik olarak analiz etme ihtiyacı da artmıştır. Bir dokümanın işe yarar bilgi içerip içermediğini değerlendirmek için dokümanın ana fikri, anahtar kelimeleri ya da kavramları biliniyor olmalıdır. Türkçe için anahtar kelime çıkarma ve ana fikir çıkarma üstüne yapılmış birkaç çalışma bulunmaktadır. Kavram çıkarma çalışmaları, birkaç yabancı dil için yapılmış olmasına rağmen kaynaklarımıza göre Türkçe için henüz böyle bir çalışma yapılmamıştır.

Bu tezde, Türkçe için kavram çıkarma sistemi ortaya konulmuştur. Türkçe karakterlerin bilgisayar diline uymaması ve Türkçenin sondan eklemeli karmaşık yapısından dolayı öncelikle bir ön işleme aşaması gereklidir. Ön işlemenin sonucunda, çekim eklerinden de ayrılmış olan kelimelerin sadece isim türünde olanları kullanılmıştır. Çoğu kavramın tanımı isim türünde kelimeleri kullanarak yapılabilir. Bunun için, benzer kelimeleri sınıflandırmanın kavram çıkarma çalışması için yararlı olabileceği düşünülmüştür. Bu istatiksel metotların ardından doğal dil işleme yöntemleri de uygulanıp test derlemindeki dokümanlar kavramlarla tanımlanmıştır. Derlem üzerinde kelime, sınıf ve kavram bazında olmak üzere çeşitli denemeler yapılmıştır. Sonuç olarak, sistem üretmesi gerekenden daha fazla kavram üretmiş olmasına rağmen, yüzde 51 başarı ile dokümanlara ait kavramları bulmuştur. Kavramların yapı itibariyle dokümanlarda aynen geçmeme ihtimali ve Türkçenin karmaşık yapısı düşünülürse bu sonuç oldukça başarılı olarak değerlendirilebilir.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

BoDis	The Boun Morphological Disambiguator
BoMorP	The Boun Morphological Parser
CES	Concept Extraction System
CRM	Customer Relationship Management
FN	False Negative
FP	False Positive
IDF	Inverse Document Frequency
KEA	Keyphrase Extraction Algorithm
LMA	Language Modeling Approaches
LSA	Latent Semantic Analysis
NLP	Natural Language Processing
POS	Part-Of-Speech
SRG	Semantic Relationship Graph
SVM	Support Vector Machines
TF	Term Frequency
TN	True Negative
TP	True Positive
UTF-8	8-Bit Unicode Transformation Format
XML	Extensible Markup Language

1. INTRODUCTION

There is a vast amount of available electronic information which is online editions of newspapers, academic journals, conference proceedings, Web sites, blogs, wikis, e-mails, instant messaging, surveys, and in scientific, government, or corporate databases. Using all these electronic information, controlling, indexing or searching is not feasible and possible for a human. For search engines, users have to know the keywords of the subject that they search, since search engines use top down approach in order to find information in textual materials. The necessity of analyzing unstructured texts automatically is apparent. Users do not have to know the query terms and the main idea of the searched documents. If the concept of a document is known, a general knowledge about it also is known.

Concept is a research area related to philosophy more than linguistics. Thus, it is useful first to look at the definition of a concept from a philosophical point of view. In philosophy, a concept is defined as a thing apprehended by human thought and concepts are elements of thoughts and facts [1]. Concepts are different from words. Words are used for naming the concepts. It is possible that a single word can correspond to more than one concept or several words can define a single concept. These relationships are related to context and scope, which are the two ingredients of a concept.

Concept extraction study aims at obtaining efficient solutions to some problems which are harder to solve using data mining. Crangle et al.[2] define concept extraction as follows:

> *"Concept extraction is the process of deriving terms from natural-language text that are considered representative of what the text is about. The terms are natural-language words and phrases which may or may not themselves appear in the original text."*

For concept extraction methods from unstructured texts there are two approaches; expert-based approach and statistical approach. Expert-based approach can be named as rule based approach. It has several disadvantages such as finding specialists on subjects and developing learning based systems. In statistical approaches, statistical methods are applied to the training data and models are built. Bayesian networks, neural networks,

support vector machines (SVM), and latent semantic analysis (LSA) are some of the statistical methods used in this area. Natural Language Processing (NLP) approach is different than these approaches in the sense that it uses the speed and cost effectiveness of the statistical approach but sometimes may require human intervention [3]. For linguistics-based approaches human intervention may be needed at the beginning to develop dictionaries for a particular industry or field of study. However, it has several considerable advantages such as getting more precise results quickly. Concepts can be extracted by using these models.

For English there are some studies done for concept extraction such as [2] and [4], and there are some commercial softwares such as SPSS PASW Text Analytics and WordStat. These softwares also support several other languages such as Arabic, Chinese, Dutch, French, German, Hindi, Italian, Persian, Portuguese, Romanian, and Russian. Moreover, there are some studies for unstructured Turkish documents for key phrase extraction such as [5] and [6]. However, key phrase extraction is different from concept extraction that key phrases are written in documents as they appear, but concepts do not have to be written in documents. There is neither study on concept extraction nor software for Turkish. In this study a concept extraction system for Turkish is proposed.

In chapter 2, literature survey about concept extraction and related works are presented. In chapter 3, the methodology in order to develop a concept extraction system for Turkish is explained. In chapter 4, experiments, their results and evaluations are given. In chapter 5, a summary of the study done and the results obtained are given.

2. LITERATURE SURVEY

Concept extraction is divided into two areas: concept recognition which aims to find all possible concepts of documents, and concept summarization which aims to select important concepts of documents [7]. Concepts can be words or phrases. Therefore, initially sentences are divided into their words and phrases. In order to divide sentences grammatical and syntactic methods are used which are tested in ontology learning [8], lexical extraction [9], and information retrieval systems [10]. In grammatical methods in order to parse sentences if shallow parsing is used, the whole sentence is converted into a grammatical tree where the leaves are noun and verb phrases. Then, noun phrases are selected as concepts [7]. In syntactic methods punctuation and conjunctions are used as divisors. Then, all phrases are regarded as concepts. This approach is also used in keyword extraction systems [11].

For concept extraction there are two important application areas which are indexing documents and categorizing documents. Moreover, it is used for evaluating open ended survey questions [12], mapping student portfolios [7], extracting synonymy from biomedical data [2], even for extracting legal cases of juridical events [13], and several other areas. The main reason of the usage of concept extraction in numerous fields is that concepts give an opportunity to enhance information retrieval systems [14-16].

2.1. Studies on Concept and Key Phrase Extraction from Unstructured Documents

Extracting key phrases of documents is related to extracting concepts of documents. In academic articles, generally, key phrases are listed after the summary which helps the reader to understand the context of documents before reading the whole document. In automatic key phrase extraction field some studies are presented. Keyphrase Extraction Algorithm (KEA) is an automatic keyhrase extraction algorithm which is proposed by Witten et al. [11].

The KEA is a supervised learning algorithm that is composed of two steps; training and extraction. In the training step, documents are trained with author-assigned key

phrases by Naïve Bayes Algorithm and a model is built. In the extraction step, key phrases are selected among candidate phrases by the model. Selecting candidate phrases consists of input cleaning, phrase identification, and case-folding and stemming steps. Feature calculation operation is normalization of the multiplication of Term Frequency (TF)*Inverse Document Frequency (IDF) value and distance from the beginning to the first occurrence of the phrase in the document. The evaluation of algorithm is done by comparing author-assigned key phrases and KEA generated key phrases of unstructured documents. As a result, only one or two phrases assigned by the KEA are correctly matched with the author-assigned key phrases.

The KEA was applied to Turkish documents by Pala and Cicekli by changing the stemmer and stop-words modules, and by adding a new feature to the algorithm [5]. The new feature added is named as relative length multiplier which is used in feature calculation. The evaluation is made in the same way and results are similar to the original KEA that is applied to English documents. Without the added part, relative length multiplier, results are worse than that of the English version.

In automatic key phrase extraction field there is also a study made by Wang et al. [17]. In this study, key phrases are extracted by using neural networks. First of all, from all the documents, phrases are selected and some features are calculated for all the phrases. These features are TF, IDF, whether the phrase occurs in the title or subtitle, and number of paragraphs that the phrase occurs in. These parameters are given to the neural network as an input. The algorithm is composed of training and test stages. In the training stage, the output phrase is tagged as key phrase or not. In the test stage, if the output is greater than 0.5 it is tagged as key phrase and it is tagged as non-key phrase otherwise. The results are evaluated by two different methods. One is the precision and recall method, the other is the subjective assessment of human readers. According to the first method, the algorithm is 30 per cent successful; according to the subjective assessment, the algorithm is 65 per cent successful.

According to [18], key phrases can be used for summarizing, indexing and easing search. In this study, there are two algorithms used in order to extract key phrases from documents; one of them is C4.5 [19] and the other is GenEx algorithm. Both algorithms

are supervised learning algorithms. First of all, all possible phrases are extracted from the document. The stems of the phrases are obtained by using the Potter [20] and Lovins [21] stemming algorithms. For both algorithms C4.5 and GenEx, the parameters used in them are selected from 110 distinct features of documents. The frequency of key phrases, the first occurrence of the stemmed phrase, and the information whether the phrase is a proper name or not are three of these features. The GenEx algorithm is a combination of Genitor [22] and Extractor algorithms. The Genitor algorithm is used only in the training step. After determining the best values of parameters in the training step, only the Extractor algorithm is used for the testing step. The algorithms are tested for five different corpora. For accuracy test, the precision and recall method is used. As a result, it is thought that by changing the C4.5 algorithm a little, better results might be obtained. It is seen that the GenEx algorithm performs better than C4.5. The overall success result is very low.

Identifying wheather electronic books in a digital library are useful or not is very difficult for a person. Rohini presented a study that extracts key phrases from electronic books [23] by using Language Modeling Approaches (LMA) which is proposed by Tomokiyo and Hurst [24]. According to Tomokiyo and Hurst, there are two important factors for extracting key phrases which are phraseness and informativeness. The phraseness property tests if words that appear together constitute a phrase or not. The informativeness property tests if the phrase gives information about the document or not [24]. First of all, all words in the document are separated according to an n-gram model. Rohini selected n as three, so all possible three word sequences are generated. Then, the first factor is tested that for each word in the phrase how much information is lost by assuming words as a phrase. The second factor is tested that how much information is lost by assuming the phrase is obtained from the corpus instead of the document. These factors are applied for all phrases generated and the results of the factors are summed for each phrase in order to get a score. 10 phrases with the highest scores are given as key phrases.

Key phrases in a document reflect the main idea of the document [6]. Kalaycılar and Cicekli proposed an algorithm called TurKeyX for Turkish in order to extract key phrases of Turkish documents automatically. This algorithm is an unsupervised learning algorithm that is based on statistical evaluation of noun phrases in a document. In the first step of this algorithm, the document is separated to its words and all possible phrases are listed. This

step is not successful enough that some phrases contain 17 words in them. For all possible phrases some features are tested. These are frequency of phrases with their morphemes, frequency of phrases without their morphemes, number of words in a phrase, first occurrence of a phrase, and first occurrence of the head noun of a phrase. After that, a formulation which is found by experiments by using these features is used. For each possible phrase a score is calculated by this formula. In the next step, incorrectly extracted and duplicate phrases are filtered. If a phrase is involved in another phrase, the phrase with the low score is eliminated. And only noun phrases are selected. After these operations, the phrases are sorted according to their scores. According to the length of document, 5 or 10 phrases with the highest scores are given as output. Two corpora are used in the testing process that one of them is the corpus which is used by Pala and Cicekli [5]. For this corpus both algorithms generated nearly same the results. The general success rate of TurKeyX is 25-30 per cent.

A study about extracting concepts automatically from plain texts is done by Gelfand et al. [4]. The aim of this study is grouping the related words and extracting concepts of the documents by identifying the relationships between words in documents based on a lexical database (WordNet). A directed graph called Semantic Relationship Graph (SRG) is created by using the word relationships. First of all, there is a base word list which contains some words that exist in the document. Each time, a word is taken from the list, and hypernyms and hyponyms of this word are found. If any hypernym or hyponym of this word is in the list, all the generated words are attached to the graph and to the list until a threshold value. These steps are repeated for all words in the list. Words in the list that do not add significant meanings to the document are eliminated. In other words, if words do not connect with many words in the graph, they are eliminated. In the testing step, 50-400 training examples are taken randomly from Mitchell's webkb dataset. The same set is used to train a Bayesian classifier also. The accuracy of the SRG-based classifier is significantly better than that of Naive Bayes, but also the run time was very high to create the classifier. If the base word list is created by a human specialist instead of a random list, the result gets better. In the article, the general performance of the study is not presented.

2.2. Commercial Software on Concept Extraction Subject

Several studies are done on automatic key phrase and concept extraction from unstructured documents; however unfortunately the success rate is still very low which is about 30 per cent. The most successful program in this area is PASW Text Analytics program generated by SPSS Inc.. This program runs for seven native languages which are English, French, Spanish, Dutch, German, Italian, and Portuguese [3]. Moreover, for 14 languages translations are available in the English language extractor through the use of Language Weaver Software. These languages are Arabic, Chinese, Dutch, French, German, Hindi, Italian, Persian, Portuguese, Romanian, Russian, Somali, Spanish, and Swedish. However, there is no support or any program for Turkish yet.

There is a vast amount of available electronic information. As stated before, using all these electronic information controlling, indexing or searching is not possible for a human. Text Analytics is different from searching [3]. For search engines, users have to know the keywords of the subject that they search, since search engines use a top down approach in order to find information in textual materials. On the other hand, Text Analytics uses a bottom up approach that users do not have to know the query terms. Text Analytics extracts the concepts and the main idea of documents and gives relationships between them.

SPSS Text Analytics approaches the concept extraction process as a whole and both before and after concept extraction, it has several steps. These are; preparing the text for analysis, extracting concepts, uncovering opinions, relationships, facts, and events through Text Link Analysis, building categories, building text analytics models, merging text analytics models with other data models, and deploying results to predictive models. In the technical report of Text Analytics five of these steps are explained [3].

(i) Preparing the text for analysis:

Before starting text analysis a corpus is needed. SPSS Text Analytics as mentioned above supports many different languages and file formats. First of all, for the corpus which contains different languages in it, languages of documents are recognized by an n-gram method. Document formats can be a database format or XML

based format. All different formats of documents are converted to plain text format and graphics are removed. After that, texts are separated to their paragraphs and sentences.

(ii) Extracting concepts:

The concept extraction process is realized in five major steps. The first of these is managing linguistic resources. Linguistic resources are arranged in a hierarchy. At the highest level there are libraries, compiled resources and some advanced resources. Moreover, for English, there are specialized templates for some specific application areas like CRM, gen ontology, market intelligence, genomics, IT and security intelligence. Libraries contain several types of dictionaries. There are two types of dictionaries: compiled dictionaries which end users cannot modify and other dictionaries which end users can modify. The compiled dictionaries consist of lists of base forms with part-of-speech (POS) and lists of proper names like organizations, people, locations and product names. Dictionaries which can be modified by users are type, exclusion, synonym, keyword, and global dictionaries. After that, candidate terms are extracted. Candidate terms are words or word groups which are used to define concepts of the documents. For that, linguistic and non-linguistic extraction techniques are used. After that by using named entities and the dictionaries, types are assigned to the candidate terms in order to ease the understanding of the content of the documents. In the next step, equal classes are found and merged, mistyped characters are found and corrected. Finally, all documents in the corpus are presented as indexed.

(iii)Uncovering opinions, relationships, facts, and events through Text Link Analysis:

In order to explain events and facts, Text Link Analysis helps the analysts to identify responses as positive or negative. By this capability of Text Link Analysis, connections between organizations, events and facts are revealed. These can help market intelligence, fraud detection, and life sciences research. NLP-based Text Analytics can determine structures which are written differently but have the same meaning.

(iv)Building categories:

Categorizing documents is the next step of Text Analytics. Since each dataset is different from the others, the method selection and application process can differ

according to the project, and the researcher. However, for all cases a researcher applies the methods, evaluates the results, makes changes on the method or categories, and purifies the results. SPSS Text Analytics includes automated linguistics-based methods which are concept derivation, concept inclusion, semantic networks, and co-occurrence rules. Users can choose methods to be used in the program, after categories are created they can add, remove or merge categories, and arrange elements in them.

(v) Deploying results to predictive models:

The results can be converted to predictive models automatically. In the implementation phase of Text Analytics, evaluating results and combining with models are possible. By using models, users can for example, generate sales offer, identify creditworthy customers, highlight positive or negative customers, or suggest patterns of possible criminal behavior.

Another area in which text analysis is used frequently is survey analysis. SPSS generated a tool for this aim that is PASW Text Analytics for Survey [12]. In surveys, close-ended questions are not enough to interpret results correctly since responses to questions frame and limit possible answers. In order to obtain comprehensive and correct information from surveys, open-ended questions have to be asked. The words that respondents choose are even important while interpreting the surveys. The approach of this tool is the same as SPSS PASW Text Analytics which is specialized for surveys.

Another tool generated in this research area is WordStat. WordStat is a tool which is generated in order to extract information from documents, feedbacks of customers, interview transcripts or open-ended responses [25]. Usage areas of it are listed in its manual like below:

- Content analysis of open-ended responses.
- Business intelligence and competitive analysis of web sites.
- Information extraction and knowledge discovery from incident reports, customer complaints, and messages.
- Analysis of news coverage or scientific literature.
- Automatic tagging and classification of documents.
- Taxonomy development and validation.

- Fraud detection, authorship attribution, patent analysis

The main properties of WordStat are integrated text mining analysis, visualization tools, hierarchical categorization dictionary, word patterns, phrases and proximity rules, vocabulary and phrase finder for extraction of technical terms, recurring ideas and themes, keyword-in-context and keyword retrieval tools for easy identification of relevant text segments, machine learning algorithm for automatic document classification (Naive Bayes and K-Nearest Neighbors) with automatic features selection and validation tools, and importation of documents and exportation of data, tables and graphs. In the program, there are some words and their categories are stored. Users can load categories and exclusion files, add or remove categories, and add several rules for each analysis. The program is available for English, French, Italian and Spanish. After classifying documents, some statistics are accessible like term frequency and TF*IDF; statistical calculations can be done between words and documents like Chi-square, Student's F, Tau, and Somers' D; the relationship between documents and categories can be presented by tools like dendrogram, heat map, and proximity plot.

3. THE METHODOLOGY

3.1. Corpus and Pre-processing

In order to develop a Concept Extraction System (CES) for Turkish, a corpus has to be determined to work on. The first step in this work is finding comprehensive Turkish documents. Then pre-processing processes start. In order to run codes on documents, they all have to be converted to txt format. Txt files have to be saved in 8-bit Unicode Transformation Format (UTF-8).

UTF-8 format is a Unicode transformation format with an octet (8 bit) [26]. It encodes Unicode characters lossless such that each Unicode character is 1 to 4 octets, where the number of octets depends on the integer value assigned to the Unicode character. It represents each character in the range U+0000 through U+007F as a single octet.

While saving documents in UTF-8 format, the characters that cannot be represented in UTF-8 format are also eliminated. Then they are prepared for the programs used next. All documents in the corpus are tokenized that a blank character is inserted before punctuation characters.

3.2. Operating on Words and Creating Nouns List

Concepts can be determined by nouns and noun phrases. Therefore, in order to obtain concepts of documents, nouns of documents have to be extracted. Extracting nouns of documents and eliminating inflectional morphemes are difficult issues for Turkish. In this process, The Boun Morphological Parser (BoMorP) and The Boun Morphological Disambiguator (BoDis) programs [27] are used. They parse documents nearly perfectly that the accuracy is 97 per cent.

The BoMorP is a state-of-the-art finite-state transducer-based implementation of Turkish morphology [27]. The program takes documents as input whose sentences have

been tokenized as explained above. It parses the words and identifies their roots, inflectional morphemes and derivational morphemes. For Turkish, usually there are alternative parses for a word. The BoMorP gives all possibilities of roots and morphemes of words as output. The POS of the root and the morphemes are represented in square brackets. If the morpheme is a derivational morpheme '-' sign is put before it. If the morpheme is an inflectional morpheme '+' sign is put before it. For example, for the word 'alın' the output of morphological parser is as follows:

alın[Noun]+[A3sg]+[Pnon]+[Nom]
al[Noun]+[A3sg]+Hn[P2sg]+[Nom]
al[Adj]-[Noun]+[A3sg]+Hn[P2sg]+[Nom]
al[Noun]+[A3sg]+[Pnon]+NHn[Gen]
al[Adj]-[Noun]+[A3sg]+[Pnon]+NHn[Gen]
alın[Verb]+[Pos]+[Imp]+[A2sg]
al[Verb]+[Pos]+[Imp]+YHn[A2pl]
al[Verb]-Hn[Verb+Pass]+[Pos]+[Imp]+[A2sg]

As seen a word can be separated to its morphemes in many ways. In order to solve this problem, the BoDis program is used [27, 28]. In this program, Sak calculates a ratio for possible roots and morphemes according to the document content. The averaged perceptron algorithm is applied to re-rank the n-best candidate list. The accuracy of the disambiguator program is 97.81 which is the highest recorded accuracy for Turkish. The outputs of the BoMorP program are the inputs of the BoDis program. The BoMorP and the BoDis programs are applied to all the documents in the corpus.

In order to exemplify, there is a sentence below from a document:
"Yapı üretim süreci ardışık karakterdeki alt üretim süreçlerinden oluşmaktadır."

After applying the BoMorP to the sentence, its outcome is given in Table 3.1.

Table 3.1. Sample output of the BoMorP program

yapı yapı[Noun]+[A3sg]+[Pnon]+[Nom]

Table 3.1. Sample output of the BoMorP program (contd.)

```
üretim
üretim[Noun]+[A3sg]+[Pnon]+[Nom]

süreci
süreç[Noun]+[A3sg]+SH[P3sg]+[Nom]
süreç[Noun]+[A3sg]+[Pnon]+YH[Acc]
süre[Noun]+[A3sg]+[Pnon]+[Nom]-CH[Noun+Agt]+[A3sg]+[Pnon]+[Nom]

ardışık
ardışık[Adj]

karakterdeki
karakter[Noun]+[A3sg]+[Pnon]+DA[Loc]-ki[Adj+Rel]

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

üretim
üretim[Noun]+[A3sg]+[Pnon]+[Nom]

süreçlerinden
süreç[Noun]+[A3sg]+lArH[P3pl]+NDAn[Abl]
süreç[Noun]+lAr[A3pl]+SH[P3sg]+NDAn[Abl]
süreç[Noun]+lAr[A3pl]+SH[P3pl]+NDAn[Abl]
süreç[Noun]+lAr[A3pl]+Hn[P2sg]+NDAn[Abl]

oluşmaktadır
oluş[Verb]+[Pos]-mAk[Noun+Inf1+A3sg+Pnon]+DA[Loc]-
DHr[Verb+Pres+Cop]+[A3sg]
oluş[Verb]+[Pos]-mAk[Noun+Inf1+A3sg+Pnon]+DA[Verb+Loc]-
DHr[Verb+Pres+Cop]+[A3sg]
oluş[Verb]+[Pos]+mAktA[Prog2]+[A3sg]+DHr[Cop]+[A3sg]

. .[Punc]
```

After applying the BoDis, its outcome is given in Table 3.2.

Table 3.2. Sample output of the BoDis program

```
Yapı
yapı[Noun]+[A3sg]+[Pnon]+[Nom] : 8.486328125

üretim
üretim[Noun]+[A3sg]+[Pnon]+[Nom] : 8.5498046875

süreci
süreç[Noun]+[A3sg]+[Pnon]+YH[Acc] : 12.4443359375
süreç[Noun]+[A3sg]+SH[P3sg]+[Nom] : 12.013671875
süre[Noun]+[A3sg]+[Pnon]+[Nom]-CH[Noun+Agt]+[A3sg]+[Pnon]+[Nom] :

9.939453125
```

Table 3.2. Sample output of the BoDis program (contd.)

ardışık
ardışık[Adj] : 16.23828125

karakterdeki
karakter[Noun]+[A3sg]+[Pnon]+DA[Loc]-ki[Adj+Rel] : 15.0732421875

alt
alt[Noun]+[A3sg]+[Pnon]+[Nom] : 11.7607421875
alt[Adj] : 7.8544921875

üretim
üretim[Noun]+[A3sg]+[Pnon]+[Nom] : 8.5498046875

süreçlerinden
süreç[Noun]+lAr[A3pl]+Hn[P2sg]+NDAn[Abl] : 21.021484375
süreç[Noun]+lAr[A3pl]+SH[P3sg]+NDAn[Abl] : 15.6875
süreç[Noun]+lAr[A3pl]+SH[P3pl]+NDAn[Abl] : 15.609375
süreç[Noun]+[A3sg]+lArH[P3pl]+NDAn[Abl] : 13.904296875

oluşmaktadır
oluş[Verb]+[Pos]-mAk[Noun+Inf1+A3sg+Pnon]+DA[Verb+Loc]-
DHr[Verb+Pres+Cop]+[A3sg] : 27.03125
oluş[Verb]+[Pos]-mAk[Noun+Inf1+A3sg+Pnon]+DA[Loc]-
DHr[Verb+Pres+Cop]+[A3sg] : 18.712890625
oluş[Verb]+[Pos]+mAktA[Prog2]+[A3sg]+DHr[Cop]+[A3sg] : 16.365234375

. .[Punc] : 16.125

After the disambiguation process, the nouns in the documents are selected. If the highest probability of root of the word is noun, it is selected unless it is acronym, abbreviation, or proper name. These POS tags are also represented as noun in the root square bracket, but in the next square bracket their original POS is written. So, the second square bracket is also checked in order to obtain the correct nouns list. Abbreviation and acronyms are shortened forms of words or phrases but they cannot determine the main idea of documents alone. Proper names like person names and country names etc. cannot be the meaning of a document. Therefore, they are eliminated. There are some samples below whose roots are nouns but their specified POS tags are different.

cm cm[Noun]+[Abbr]+[A3sg]+[Pnon]+[Nom]

ISO ISO[Noun]+[Acro]+[A3sg]+[Pnon]+[Nom]

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

In the output of the BoDis program, letters are also represented as nouns. They are mostly acronym or abbreviation as the sample below, but sometimes they are only listed as nouns. The letters which are in this format are eliminated from the nouns list. Finally, the list is controlled by a human specialist manually. Nouns which contain two letters are also eliminated if they are meaningless.

g g[Noun]+[Abbr]+[A3sg]+[Pnon]+[Nom]
G[Noun]+[Acro]+[A3sg]+[Pnon]+[Nom]

Inflectional morphemes are removed from nouns. Therefore, for example, the root form of all “sistem, sistemler, sistemlerin, sistemde, sistemin, sisteme, etc.” is regarded as “sistem” and their frequencies are added to the “sistem” noun. However, derivational morphemes are kept as they appear. For example, the noun “çözüm” is derived from the verb “çöz”, however the noun “çözüm” noun is added to the nouns list in this form. All nouns are listed for the documents and their frequencies are calculated. Then all nouns of the documents are gathered in one file, the same words in the documents are merged and their frequencies are added. Moreover, the nouns which occur in the documents rarely are considered as they cannot give the main idea of them. If the frequencies of the nouns are less than three, they are eliminated in order to decrease the size of the list and speed up later processing.

3.3. Clustering Cumulative Nouns List

Concepts can be defined by nouns. Therefore, clustering similar nouns can be helpful in order to determine concepts. In order to cluster words, some clustering methods are applied to the cumulative nouns list which are hierarchical clustering and k-means clustering. These clustering methods are unsupervised learning algorithms which do not need any training step to pre-define the categories and label the documents. So, there is no need for a training set while applying the algorithms.

3.3.1. Hierarchical Clustering

The hierarchical clustering method clusters similar instances in a group by using similarities of them [29]. This needs the use of a similarity measure which is generally Euclidean measure. Therefore a similarity matrix of instances has to be created before running the method. Hierarchical clustering can be categorized into two; agglomerative (bottom-up) and divisive (top-down) clustering.

3.3.1.1. Agglomerative Clustering. An agglomerative clustering algorithm starts with clusters which each of them contains only one instance and at each iteration merges the most similar clusters until the stopping criterion is met such as a requested number *k* of clusters is achieved [29, 30]. The algorithm of agglomerative clustering [31]:

(i) Start by assigning each item to its own cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters equal the distances (similarities) between the items they contain.
(ii) Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one less cluster.
(iii) Compute distances (similarities) between the new cluster and each of the old clusters.
(iv) Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

At third step, the distance (or similarity) matrix is updated after merging two items. This update can be done by three different approaches:

Single-link clustering: The distance between two clusters is defined as the smallest distance from any member of one cluster to any member of the other cluster [29, 31].

$$d(G_i, G_j) = \min_{x^r \in G_i, x^s \in G_j} d(x^r, x^s) \tag{3.1}$$

Similarity matrix is exact opposite of distance matrix. In other words, while creating similarity matrix, we consider the similarity between two clusters is equal to the biggest value from any member of one cluster to any member of the other cluster [30, 31].

$$s(G_i, G_j) = \max_{x^r \in G_i, x^s \in G_j} s(x^r, x^s) \quad (3.2)$$

Complete-link clustering: The distance between two clusters is defined as the largest distance from any member of one cluster to any member of the other cluster [29, 31].

$$d(G_i, G_j) = \max_{x^r \in G_i, x^s \in G_j} d(x^r, x^s) \quad (3.3)$$

Similarity matrix is exact opposite of distance matrix. In other words, while creating similarity matrix, we consider the similarity between two clusters is equal to the smallest value from any member of one cluster to any member of the other cluster [30, 31].

$$s(G_i, G_j) = \min_{x^r \in G_i, x^s \in G_j} s(x^r, x^s) \quad (3.4)$$

Average-link clustering: The distance or the similarity between two clusters is defined as the average distance or similarity from any member of one cluster to any member of the other cluster [29, 30, 31].

$$a(G_i, G_j) = \operatorname{avg}_{x^r \in G_i, x^s \in G_j} a(x^r, x^s) \quad (3.5)$$

3.3.1.2. Divisive Clustering. A divisive algorithm can be considered as the reverse form of an agglomerative algorithm that starts with one cluster containing all instances and at each iteration split the most appropriate cluster until a stopping criterion such as a requested number *k* of clusters is achieved [29, 30]. The algorithm:

(i) Start with one cluster containing all items
(ii) For each iteration split the cluster into two from the furthest (or dissimilar) item
(iii) Assign rest of the documents to one of the new clusters according to closeness (or similarity) of items.
(iv) Repeat steps 2 and 3 until all items are clustered into N clusters of size 1.

3.3.2. K-means Clustering

In k-means clustering, first of all, the means of k clusters are selected randomly. Then all points in the sample set are assigned to the cluster that is nearest to them. Then all means of k clusters are calculated again with new points added them, until values of means do not change. In Alpaydin [29], the pseudo-code of this algorithm is given as in Figure 3.1 where $\boldsymbol{m}$ is sequence of means, $\boldsymbol{x}$ is sequence of samples, and b is sequence of estimated labels.

Initialize $\boldsymbol{m}_i$, $i = 1,\ldots,k$, for example, to k random $\boldsymbol{x}^t$

Repeat

For all $\boldsymbol{x}^t \in X$

$b_i^t \leftarrow 1$ if $\| \boldsymbol{x}^t - \boldsymbol{m}_i \| = \min_j \| \boldsymbol{x}^t - \boldsymbol{m}_j \|$

$b_i^t \leftarrow 0$ otherwise

For all $\boldsymbol{m}_i$, $i = 1,\ldots,k$

$\boldsymbol{m}_i \leftarrow \sum_t b_i^t \, x^t \; / \sum_t b_i^t$

Until $\boldsymbol{m}_i$ converge

Figure 3.1. K-means algorithm, taken from [29], p.149.

3.4. Application of Clustering Algorithms

First of all, document-noun matrix is created from merged bag-of-words, which holds documents in rows, nouns in columns, and the intersection of a row and a column gives the number of that noun contained in the document. A sample of the matrix created from the corpus used in this work is given in Appendix A.1.

Hierarchical clustering algorithms are coded in MATLAB. Firstly, document-noun matrix is converted to a similarity matrix with cosine similarity. For agglomerative clustering, at each step the most similar items are found and merged. Then, similarity

matrix is updated and these are repeated until reaching the determined number of clusters. The method of updating similarity matrix determines whether it is single, complete or average link clustering. Update function is rewritten for these methods. For divisive clustering, for each step the least similar items are found and split. These are repeated until reaching the determined number of clusters. The results of hierarchical clustering algorithms are not good enough that it clusters most words in just a cluster. Cluster count is changed such as 25, 50 and 100 but the results do not change.

For k-means clustering algorithm, Tanagra program is used because of its clear and good visual appearance. Tanagra is a free, open source data mining software for academic and research purposes [32]. It allows other researchers to add their own data mining methods. The design of its GUI is easy to use. A project is created by adding data mining file to the project. Document-noun matrix is used to create a project. Then k-means clustering algorithm is run by changing parameters such as changing numbers of clusters to 10, 50, 75 and 100. Other parameters like maximum iteration and trials do not affect the results. The best results are obtained for cluster number 100. Clusters are assessed by human specialists. It is seen that the k-means algorithm performs much better than hierarchical clustering algorithms.

3.5. Assigning Clusters to Documents

After clustering operation, the clusters are assigned to the documents. This operation is done by searching the nouns of the documents in the words of the clusters. A ratio is calculated for each possible cluster of the documents. The ratio of the possible cluster of the document is calculated by dividing number of the words in the possible cluster of the document to the number of words in that cluster. If the ratio is more than a threshold value, the cluster is assigned to the document. So, it can be said that this document can be defined by this cluster. The threshold is selected as “1”, in other words, if a document contains all words of a cluster, this cluster is assigned to that document, because it is seen that if a document is related to a cluster it contains all words of that cluster. More than one cluster can be assigned to a document. A cluster can be assigned to more than one document also. The pseudo-code that implements this algorithm is given in Figure 3.2.

3.6. Identifying Documents by Concepts

The main aim of this study is defining documents with concepts. Therefore, a transition has to be done from words and clusters to concepts. In concept extraction programs like SPSS PASW Text Analytics and WordStat, dictionaries are used in order to identify documents by concepts [3, 25]. As explained in Section 2, these dictionaries

Algorithm: Assigning Clusters to Documents

Input

F1: Documents-Nouns file

F2: Clusters-Words file

Output

F3: Documents-Clusters file

Begin

1: *L1* <- Read *F1* to list

2: *L2* <- Read *F2* to list

3: **for each** word *w* in *L1*

4: Search cluster *cl* of *w* in *L2*

5: Append *cl* to *L1*

6: **end for**

7: **for each** document *d*

8: *L3* <- Read clusters of *d* in *L1*

9: *L4* <- Read words of *d* in *L1*

10: **for each** cluster *cl* in *L3*

11: *A* <- Calculate count of words of *cl* in *L4*

12: *B* <- Calculate count of words of *cl*

13: **if (***A/B >= Threshold***)**

14: Write *d* + *cl* to *F3*

15: **end if**

16: **end for**

17: **end for**

End

Figure 3.2. Pseudo-code of assigning clusters to documents

consist of concepts and words related to these concepts. In both programs, users can add or remove concept categories or words to the categories. Like these programs it is decided to create concept categories and words related to them. So, concepts have to be assigned to clusters according to words they contain by human specialists. Then, concepts are assigned to the documents according to their assigned clusters. The pseudo-code of assignment of the concepts to the documents is given in Figure 3.3.

Algorithm: Assigning Concepts to Documents

```
Input
        F1: Documents-Clusters file
        F2: Clusters-Concepts file
Output
        F3: Documents-Concepts-Count file
Begin
1:      L1 <- Read F1 to list
2:      L2 <- Read F2 to list
3:      for each document i
4:              L3 <- Read clusters of i
5:              L4 <- empty
6:              for each cluster cl in L3
7:                      L5 <- read concepts of cl
8:                      for each concept c in L5
9:                              if (L4 does not contain c)
10:                                     Add c + "1" to L4
11:                             else
12:                                     Increase count of c in L4
13:                             end if
14:                     end for
15:             end for
16:             Write L4 to F3
17:     end for
End
```

Figure 3.3. Pseudo-code of assigning concepts to documents

3.7. Illustration of the Methodology

Explaining the methodology with a hypothetical example will lead it to understand better. Let there be five documents in the corpus, and their key files which contain the keywords of the documents. The key files are used in testing step but some processes are also applied to them to prepare for testing. These documents are saved according to UTF-8 standards. After pre-precessing step, the BoMorP and BoDis are applied and nouns of the documents are extracted. As a result of this step, the document numbers and their nouns are listed as shown in Table 3.3. Same pre-processing step is also applied to the key files of the documents. Table 3.4 shows the number of the key files of the documents and their nouns.

Table 3.3. Documents and their nouns

docno	**nouns**
d1	yapı, sistem, entegrasyon, yapı
d2	sistem, yaklaşım, sistem
d3	yapı, malzeme, yangın, sınıf
d4	özellik, geometri, alaşım
d5	betonarme, yapı, analiz, yapı

Table 3.4. Key files and their nouns

key file no	**nouns**
1	yapı, bina
2	sistem, analiz
3	malzeme, yangın
4	sistem
5	analiz

Before applying clustering algorithms, a document-noun matrix is created as in Table 3.5.

Several algorithms are applied to the document-noun matrix. The algorithm which fits the dataset best is k-means algorithm by Tanagra. The best *k* cluster number is found

as seven empirically. As a result of this program, the cluster numbers and their members are given in Table 3.6.

Table 3.5. Document-noun matrix

doc no	yapı	sis tem	enteg rasyon	yakla şım	malze me	yan gın	sı nıf	özel lik	geo metri	ala şım	beto narme	ana liz
Doc1	2	1	1	0	0	0	0	0	0	0	0	0
Doc2	0	2	0	1	0	0	0	0	0	0	0	0
Doc3	1	0	0	0	1	1	1	0	0	0	0	0
Doc4	0	0	0	0	0	0	0	1	1	1	0	0
Doc5	2	0	0	0	0	0	0	0	0	0	1	1

Table 3.6. Clusters and their members

cluster no	**members**
1	betonarme, analiz
2	malzeme, yangın, sınıf
3	-
4	yapı
5	sistem, yaklaşım
6	entegrasyon
7	özellik, geometri, alaşım

After that, the clusters are assigned to the documents and key files. While assigning the clusters to the documents a threshold value is determined. This value is “1”, in other words if a document contains all words of a cluster, this cluster is assigned to that document. For the key files a threshold is not considered. If a key file contains a word of a cluster, this cluster is assigned to that key file. The document numbers and their clusters are shown in Table 3.7. As seen, although the first document contains the word “sistem” of the fifth cluster, this cluster is not assigned to the document due to the threshold value. The key file numbers and their clusters are shown in Table 3.8.

Table 3.7. Documents and their clusters

doc no	cluster no
1	4, 6
2	5
3	2, 4
4	7
5	1, 4

Table 3.8. Key files and their clusters

key file no	cluster no
1	4
2	1, 5
3	2
4	2
5	1

Table 3.9. Clusters and their concepts

cluster no	concepts
1	yapı
2	malzeme
3	-
4	yapı
5	sistem
6	sistem
7	malzeme

Finally, documents are identified by concepts via the clusters. Therefore, initially concepts are assigned to the clusters and to the key files by a human specialist according to words they contain. The specialist decided to the concepts as “yapı, malzeme, sistem” for this corpus. Then, the concepts are assigned to the documents according to the clusters which are assigned to the documents. Some concepts can be assigned to the documents more than once because of the similarity of the clusters. Table 3.9 shows the cluster

numbers and their assigned concepts. Table 3.10 shows the document numbers and their concepts. As seen in Table 3.10, the word “yapı” is assigned to the fifth document twice, because both the first and the fourth clusters which are assigned to the fifth document are related to the concept “yapı”. Table 3.11 shows the key file numbers and their assigned concepts.

Table 3.10. Documents and their concepts

doc no	**concepts**
1	yapı, sistem
2	sistem
3	malzeme, yapı
4	malzeme
5	yapı (2)

Table 3.11. Key files and their concepts

key file no	**concepts**
1	yapı
2	sistem
3	malzeme
4	sistem
5	yapı

After this process, the testing process starts.

4. EXPERIMENTS AND EVALUATIONS

4.1. Selecting Corpus

In order to develop a CES for Turkish, a corpus is needed to work on. The first step in this work is finding comprehensive Turkish documents. Turkish documents which are related to each other are searched. Online archives of Journal of The Faculty of Engineering and Architecture of Gazi University (*Gazi* corpus) [33] are selected as a corpus which is also used in [5] and [6]. It contains 60 Turkish articles and 60 .key files which contain the keywords of the articles.

4.2. Application of the Methodology

After selecting a corpus the methodology is applied to the corpus as explained before. Application of the methodology can be summarized as follows; the detailed explanation can be seen in Section 3:

(i) The articles and the key files are prepared with pre-processing procedures.
(ii) Only nouns of the articles are selected and cumulative nouns list is created.
(iii) Document-noun matrix is created. The matrix is clustered by k-means algorithm. 100 clusters are created. The 100 clusters and the words they contain are listed in Appendix A.2.
(iv) Clusters are assigned to the articles and the key files. The article numbers and the clusters assigned to them are given in Appendix B.1. The key file numbers and the clusters assigned to them are given in Appendix B.2
(v) Concepts are assigned to the key files and to the clusters. The articles are identified by concepts via the clusters assigned to the articles. The article numbers, the concepts assigned to them and count of repeated concepts are given in Appendix C.1. The key file numbers and the concepts assigned to them are given in Appendix C.2.

4.3. Testing

4.3.1. Testing Methodology

The most significant part of most projects is experiments part since the correctness of the study can be assessed in this part. Several tests are applied to the results which are obtained by applying the methodology to the corpus. These tests are test by words, test by clusters, and test by concepts. Precision and recall are used in order to give results which are widely used metrics to evaluate correctness of results of data mining projects. Venn diagrams and formulas that define precision and recall are shown in Figure 4.1, and Equations 4.1 and 4.2 respectively.

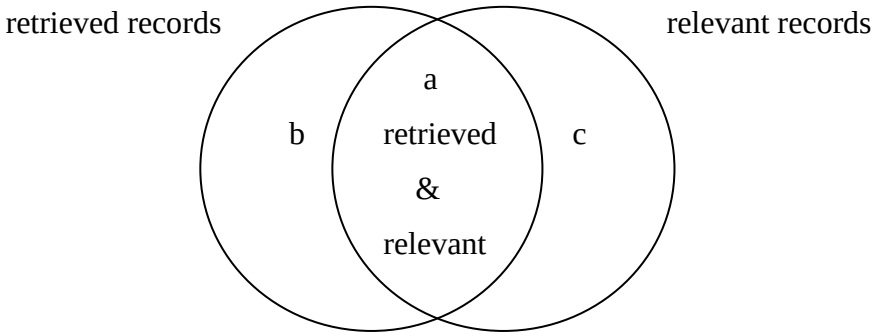

Figure 4.1. Definition of precision and recall using Venn diagrams

$$Precision = \frac{a}{a+b} \tag{4.1}$$

$$Recall = \frac{a}{a+c} \tag{4.2}$$

Table 4.1. Confusion matrix of predicted and real classes

		Predicted Class	
		Yes	No
Real Class	Yes	TP: True Positive	FN: False Negative
	No	FP: False Positive	TN: True Negative

Precision: TP / (TP+FP) (4.3)

Recall: TP / (TP + FN) (4.4)

Precision is the number of retrieved and relevant records divided by the total number of retrieved records. Recall is the number of retrieved relevant records divided by the total number of relevant records [29]. These can be formulized by using Table 4.1.

4.3.2. Test by Words

Correctness of the clusters which are assigned to the articles is tested by words via words of the key files. If the clusters are created and assigned correctly, the words of the clusters which are assigned to the articles should match with the nouns of the key files. Let us denote words of clusters which are assigned to an article as *w1*, and nouns in the key file of that article as *w2*. *w2* is searched in *w1*. For each article, the numbers of *w1*, *w2*, and the intersection of *w1* and *w2* are calculated. Then, accuracy is calculated by Equation 4.5 for each article. Here precision is not needed to be calculated because clusters contain a lot of words and limiting them is not possible in this methodology.

$$Accuracy = \frac{a}{a + c} \tag{4.5}$$

In Equation 4.5; *a* is number of the intersection of *w1* and *w2*, *(a + c)* is number of *w2*. The results of test by words are given in Table 4.2.

Average accuracy is calculated as 0.46. For this test, recall is given as accuracy. About half of the nouns of the key files are contained in the nouns of the clusters which are assigned to the articles. This information cannot explain the accuracy of the study because the clusters contain a lot of words in them; however the words of the key files are very limited. But unfortunately, although a lot of nouns are selected from the articles, only half of them are matched with the nouns of the key files. Figure 4.2 shows number of the nouns of the key files versus number of the matched nouns for each article.

Table 4.2. The results of test by words

docno	***#w1***	***#w2***	***#a***	***accuracy***
1	65	4	0	0,00
2	4	1	0	0,00
3	91	8	8	1,00
4	14	4	2	0,50
5	33	8	4	0,50
6	56	5	4	0,80
7	21	4	4	1,00
8	20	4	3	0,75
9	16	6	2	0,33
10	37	3	0	0,00
11	11	3	2	0,67
12	48	4	2	0,50
13	36	2	1	0,50
14	20	4	3	0,75
15	13	5	1	0,20
16	34	7	6	0,86
17	81	5	2	0,40
18	11	4	3	0,75
19	34	5	3	0,60
20	19	5	1	0,20
21	29	6	3	0,50
22	24	3	2	0,67
23	28	5	5	1,00
24	41	6	1	0,17
25	15	6	3	0,50
26	13	6	2	0,33
27	8	2	0	0,00
28	51	4	3	0,75
29	23	11	4	0,36
30	12	6	1	0,17
31	7	4	1	0,25
32	16	7	3	0,43
33	40	7	6	0,86
34	23	3	3	1,00
35	13	3	1	0,33
36	15	5	2	0,40
37	56	1	0	0,00
38	20	4	0	0,00
39	15	6	2	0,33
40	23	2	1	0,50
41	27	3	1	0,33
42	28	5	3	0,60
43	38	2	1	0,50
44	24	3	3	1,00
45	59	4	3	0,75
46	80	4	2	0,50
47	22	2	1	0,50
48	23	5	0	0,00
49	14	4	3	0,75
50	47	5	4	0,80
51	30	7	1	0,14
52	50	6	5	0,83
53	20	5	2	0,40
54	64	5	4	0,80
55	5	4	0	0,00
56	17	4	3	0,75
57	39	1	0	0,00
58	44	4	2	0,50
59	24	9	9	1,00
60	13	10	0	0,00

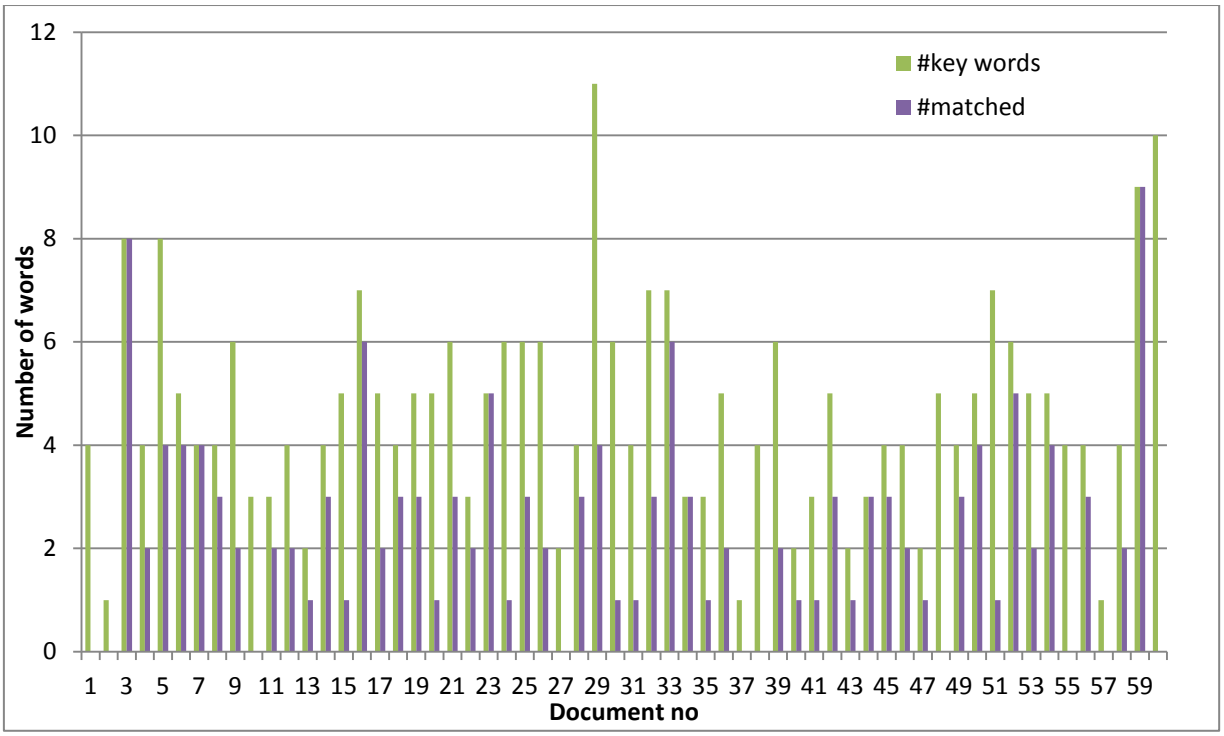

Figure 4.2. Number of the key words versus number of the matched words

4.3.3. Test by Clusters

Correctness of the clusters which are assigned to the articles is tested by clusters via the clusters of the key files. Clusters are assigned to the key files according to the nouns in them. Let us denote the clusters of an article as *cl1*, and the clusters of the key file related to that article as *cl2*. *cl1* and *cl2* are compared. For each article, the numbers of *cl1*, *cl2*, and the intersection of *cl1* and *cl2* are calculated. Then, precision and recall are calculated for each document. In Equation 4.1 and 4.2; *a* is the number of the intersection of *cl1* and *cl2*, *(a + b)* is the number of *cl1*, and *(a + c)* is the number of *cl2*. The results of test by clusters are given in Table 4.3.

Average precision and average recall are calculated as 0.50 and 0.41, respectively. As a result of the test by clusters, 41 per cent of the assigned clusters are matched with the clusters of the key files. Half of the clusters which are assigned to the articles are assigned correctly. The recall is lower than expected. Since the clusters are supposed as general topics of the articles, it shows the general topics of the articles could not be determined perfectly. However, for Turkish it can be regarded as a success because of the complexity of the language. Figure 4.3 shows number of the clusters of the key files versus number of

the matched clusters for each article. Figure 4.4 shows number of the assigned clusters versus number of the matched clusters for each article.

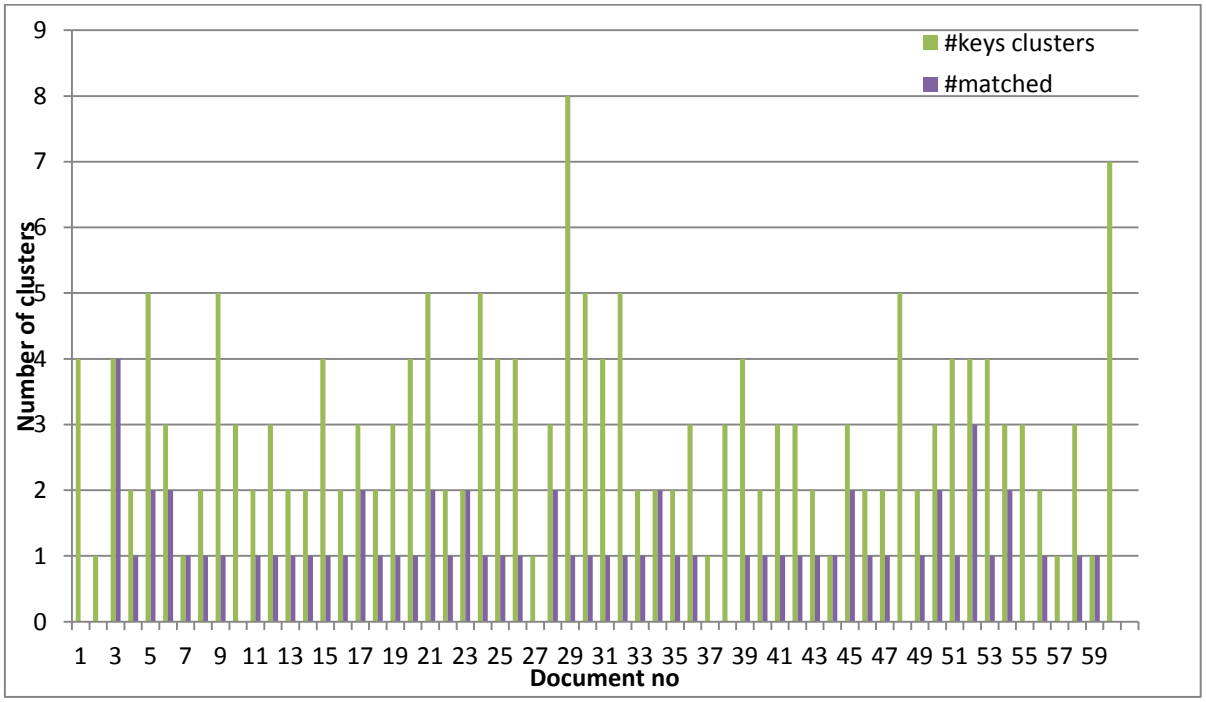

Figure 4.3. Number of the key clusters versus number of the matched clusters

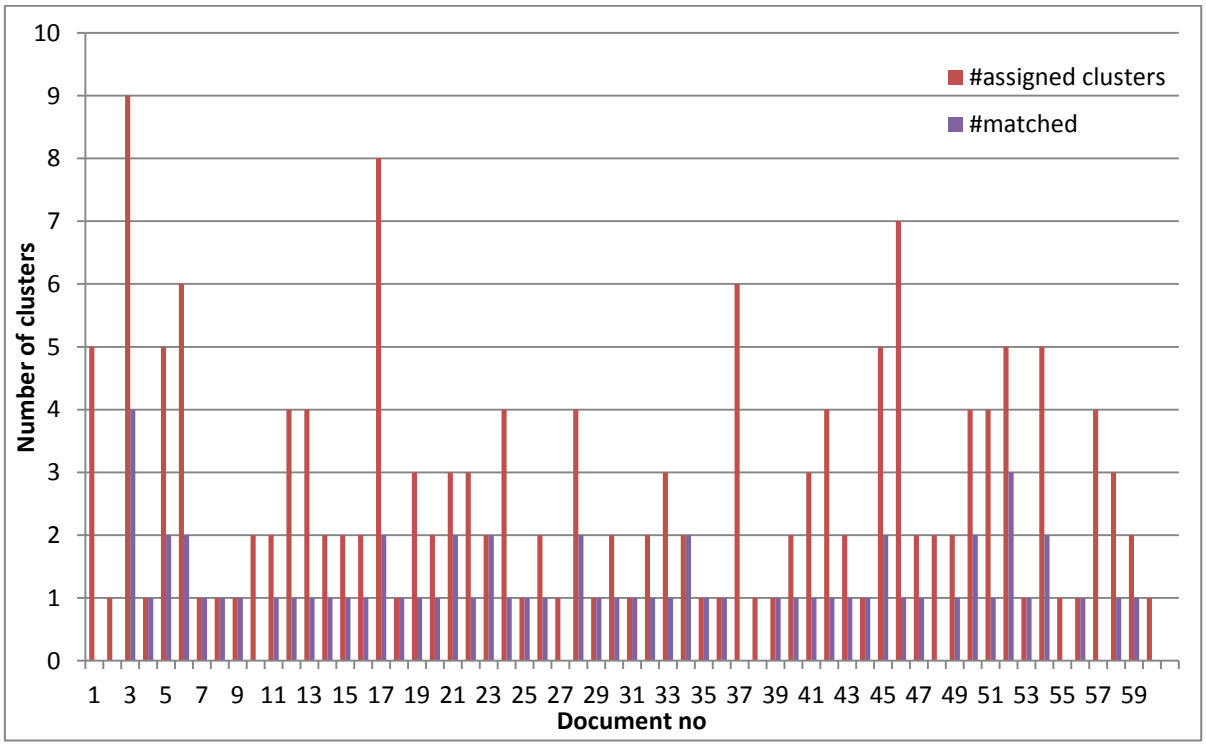

Figure 4.4. Number of the assigned clusters versus number of the matched clusters

Table 4.3. The results of test by clusters

docno	***#cl1***	***#cl2***	***#a***	***precision***	***recall***
1	5	4	0	0,00	0,00
2	1	1	0	0,00	0,00
3	9	4	4	0,44	1,00
4	1	2	1	1,00	0,50
5	5	5	2	0,40	0,40
6	6	3	2	0,33	0,67
7	1	1	1	1,00	1,00
8	1	2	1	1,00	0,50
9	1	5	1	1,00	0,20
10	2	3	0	0,00	0,00
11	2	2	1	0,50	0,50
12	4	3	1	0,25	0,33
13	4	2	1	0,25	0,50
14	2	2	1	0,50	0,50
15	2	4	1	0,50	0,25
16	2	2	1	0,50	0,50
17	8	3	2	0,25	0,67
18	1	2	1	1,00	0,50
19	3	3	1	0,33	0,33
20	2	4	1	0,50	0,25
21	3	5	2	0,67	0,40
22	3	2	1	0,33	0,50
23	2	2	2	1,00	1,00
24	4	5	1	0,25	0,20
25	1	4	1	1,00	0,25
26	2	4	1	0,50	0,25
27	1	1	0	0,00	0,00
28	4	3	2	0,50	0,67
29	1	8	1	1,00	0,13
30	2	5	1	0,50	0,20
31	1	4	1	1,00	0,25
32	2	5	1	0,50	0,20
33	3	2	1	0,33	0,50
34	2	2	2	1,00	1,00
35	1	2	1	1,00	0,50
36	1	3	1	1,00	0,33
37	6	1	0	0,00	0,00
38	1	3	0	0,00	0,00
39	1	4	1	1,00	0,25
40	2	2	1	0,50	0,50
41	3	3	1	0,33	0,33
42	4	3	1	0,25	0,33
43	2	2	1	0,50	0,50
44	1	1	1	1,00	1,00
45	5	3	2	0,40	0,67
46	7	2	1	0,14	0,50
47	2	2	1	0,50	0,50
48	2	5	0	0,00	0,00
49	2	2	1	0,50	0,50
50	4	3	2	0,50	0,67
51	4	4	1	0,25	0,25
52	5	4	3	0,60	0,75
53	1	4	1	1,00	0,25
54	5	3	2	0,40	0,67
55	1	3	0	0,00	0,00
56	1	2	1	1,00	0,50
57	4	1	0	0,00	0,00
58	3	3	1	0,33	0,33
59	2	1	1	0,50	1,00
60	1	7	0	0,00	0,00

4.3.4. Test by Concepts

Correctness of the concepts which are assigned to the articles is tested by concepts via the concepts of the key files. Let us denote the concepts which are assigned to an article as *c1*, and the concepts of the key file related to that article as *c2*. *c1* and *c2* are compared. For each article, the numbers of *c1*, *c2*, and the intersection of *c1* and *c2* are calculated. Then, precision and recall are calculated for each article. In Equations 4.1 and 4.2; *a* is the number of the intersection of *c1* and *c2*, *(a + b)* is the number of *c1*, and *(a + c)* is the number of *c2*. The results of test by concepts are given in Table 4.4.

Average precision and average recall are calculated as 0.22 and 0.51, respectively. As a result of the test by concepts, 51 per cent of the concepts which are assigned to the articles are matched with the concepts of the key files. 22 per cent of the concepts which are assigned to the articles are assigned correctly. This shows that more concepts are assigned than it must be. The recall being too high may be due to this fact. Since concepts are abstract entities, in other words they do not have to be written in the texts as they appear, assigning concept is a very difficult issue. Furthermore, Turkish is an agglutinative and complex language that studies on Turkish do not give high scores. For example, the success rate of key phrase extraction studies by [5] and [6] respectively are not passed over 30 per cent. Moreover, this study is the first study for Turkish in this subject that 0.51 and 0.22 cannot be considered as unsatisfactory. Figure 4.5 shows the number of the concepts of the key files versus the number of the matched concepts for each article. Figure 4.6 shows the number of the concepts assigned to the articles versus the number of the matched concepts for each article.

As a result of the test by concept, precision is considered as low; therefore it is thought limiting the number of the concepts assigned to the articles may be useful for the results. Due to the similarity of the clusters, some clusters contain same concepts. So, while assigning concepts to the articles via clusters, some concepts are assigned to the articles more than once (See Appendix C.1). Therefore, we performed another experiment in which a restriction is applied to the concepts of the articles such that if an article is defined by a concept more than once, the concepts that exist only once are eliminated. If

an article is defined by concepts only once, no elimination is applied. For evaluation, same formulas are applied which are explained in the test by concepts.

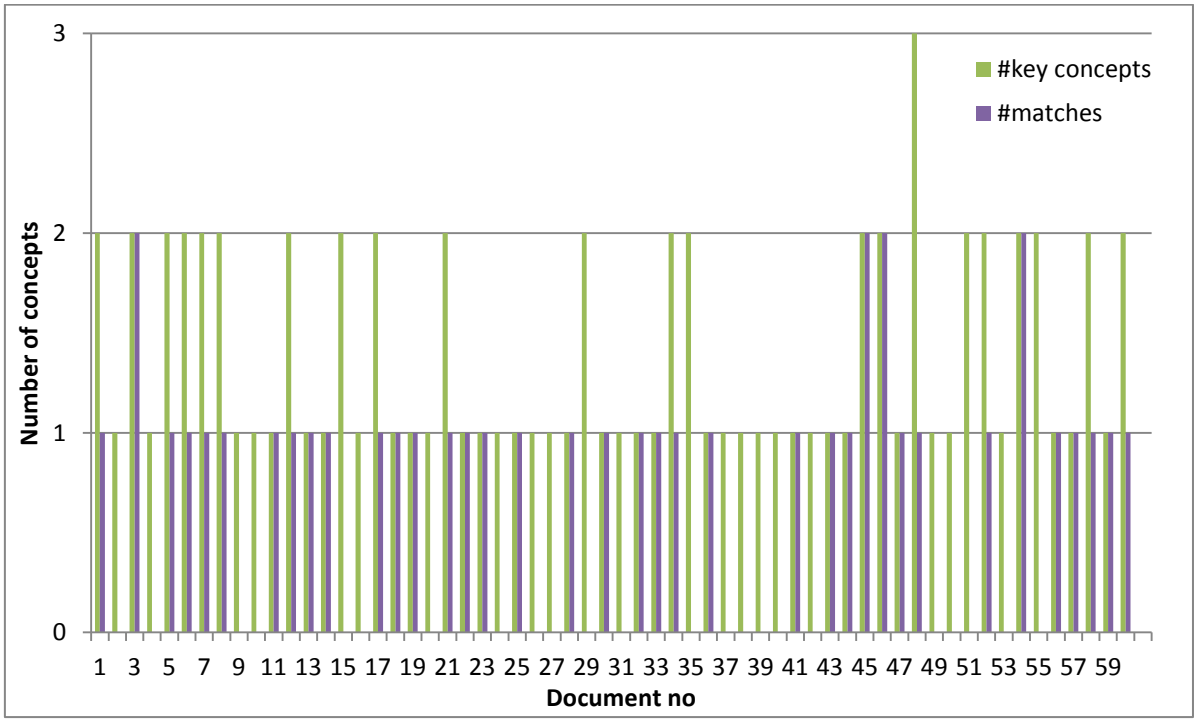

Figure 4.5. Number of the key concepts versus number of the matched concepts

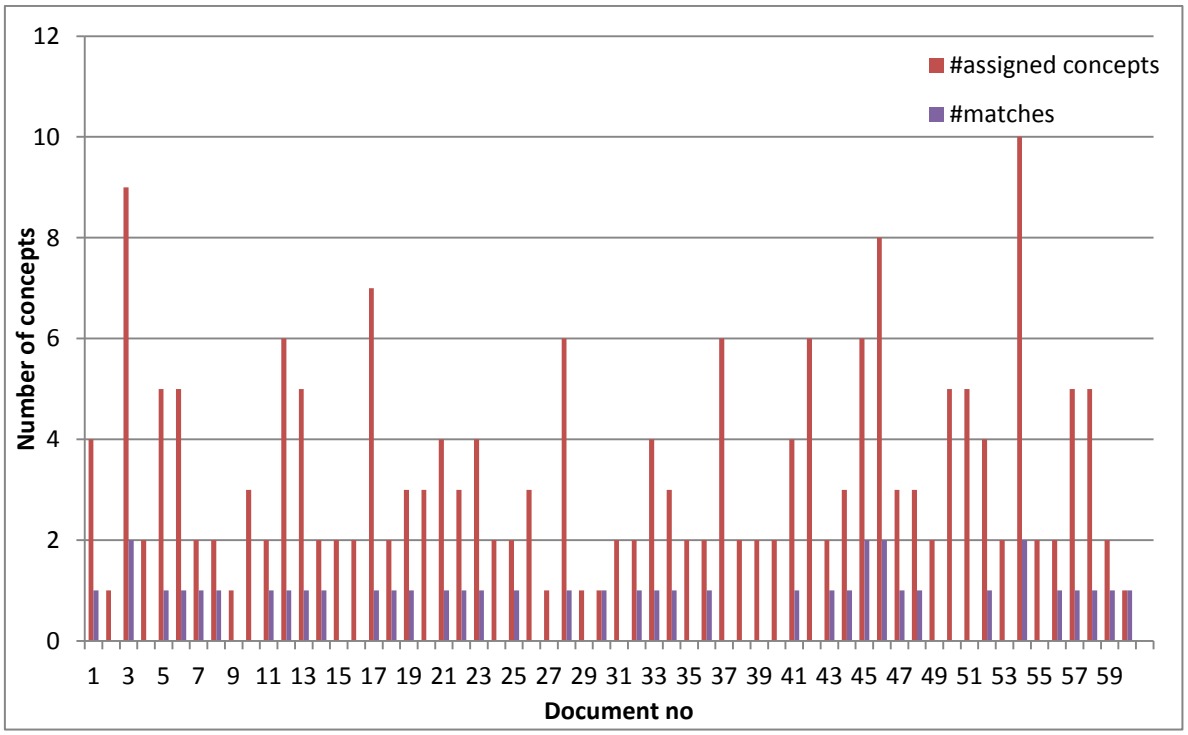

Figure 4.6. Number of the assigned concepts versus number of the matched concepts

Table 4.4. The results of test by concepts

docno	***#c1***	***#c2***	***#a***	***precision***	***recall***
1	4	2	1	0,25	0,50
2	1	1	0	0,00	0,00
3	9	2	2	0,22	1,00
4	2	1	0	0,00	0,00
5	5	2	1	0,20	0,50
6	5	2	1	0,20	0,50
7	2	2	1	0,50	0,50
8	2	2	1	0,50	0,50
9	1	1	0	0,00	0,00
10	3	1	0	0,00	0,00
11	2	1	1	0,50	1,00
12	6	2	1	0,17	0,50
13	5	1	1	0,20	1,00
14	2	1	1	0,50	1,00
15	2	2	0	0,00	0,00
16	2	1	0	0,00	0,00
17	7	2	1	0,14	0,50
18	2	1	1	0,50	1,00
19	3	1	1	0,33	1,00
20	3	1	0	0,00	0,00
21	4	2	1	0,25	0,50
22	3	1	1	0,33	1,00
23	4	1	1	0,25	1,00
24	2	1	0	0,00	0,00
25	2	1	1	0,50	1,00
26	3	1	0	0,00	0,00
27	1	1	0	0,00	0,00
28	6	1	1	0,17	1,00
29	1	2	0	0,00	0,00
30	1	1	1	1,00	1,00
31	2	1	0	0,00	0,00
32	2	1	1	0,50	1,00
33	4	1	1	0,25	1,00
34	3	2	1	0,33	0,50
35	2	2	0	0,00	0,00
36	2	1	1	0,50	1,00
37	6	1	0	0,00	0,00
38	2	1	0	0,00	0,00
39	2	1	0	0,00	0,00
40	2	1	0	0,00	0,00
41	4	1	1	0,25	1,00
42	6	1	0	0,00	0,00
43	2	1	1	0,50	1,00
44	3	1	1	0,33	1,00
45	6	2	2	0,33	1,00
46	8	2	2	0,25	1,00
47	3	1	1	0,33	1,00
48	3	3	1	0,33	0,33
49	2	1	0	0,00	0,00
50	5	1	0	0,00	0,00
51	5	2	0	0,00	0,00
52	4	2	1	0,25	0,50
53	2	1	0	0,00	0,00
54	10	2	2	0,20	1,00
55	2	2	0	0,00	0,00
56	2	1	1	0,50	1,00
57	5	1	1	0,20	1,00
58	5	2	1	0,20	0,50
59	2	1	1	0,50	1,00
60	1	2	1	1,00	0,50

Average precision and average recall are calculated as 0.16 and 0.27, respectively. Both precision and recall decrease significantly. By applying this test, precision is expected to be increased however it decreases. Moreover, recall decreases drastically that the success of assigning concepts to the articles is 27 per cent.

A restriction is again applied to the concepts of the articles which exist only once to define the articles are eliminated. Average precision and average recall are calculated as 0.04 and 0.06, respectively. Since there are a few articles which are defined by concepts more than once, the average precision and recall are too low. This shows that the results are much better without any elimination. Therefore, the result of this test can be given as 51 per cent recall with 22 per cent precision. Table 4.5 shows results of the test by concepts. A graphic which compares these results is given in Figure 4.7.

Table 4.5. Comparison of precision and recall for different number of assigned concepts

Concepts	**precision**	**recall**
no elimination	0,22	0,51
eliminate 1 if any other greater than 1 exists	0,16	0,27
eliminate 1	0,04	0,06

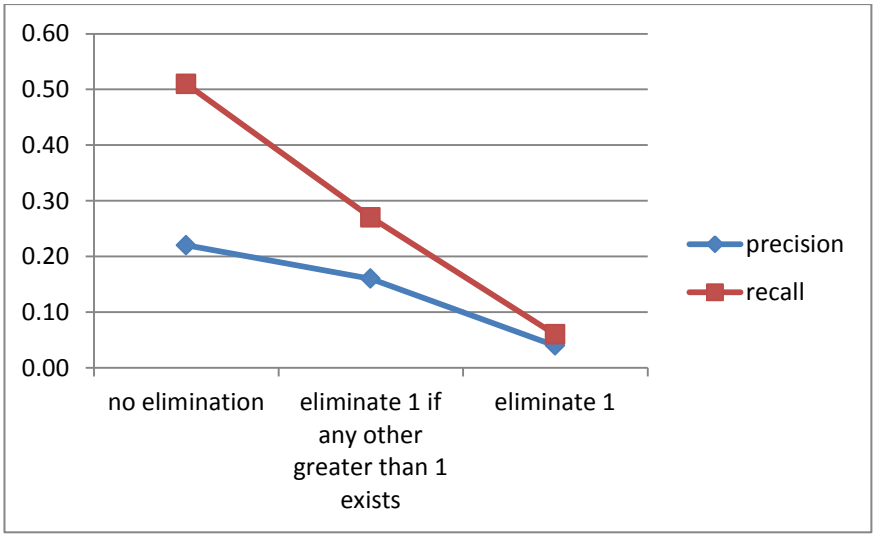

Figure 4.7. Comparison of the results of test by concepts for changing concept count

5. CONCLUSION

The growing vast amount of electronic information brings the need to analyze documents automatically to determine which documents give valuable information. Knowing the concepts of a document helps human to assess it and decide if the document is beneficial for her or not.

Concept extraction from unstructured documents is the process of extracting concepts, in other words the main idea of the texts. The main and compelling point for concept extraction is that concepts may or may not appear in the text as they are written. In this study, a concept extraction system for Turkish is proposed.

In this thesis, the methodology that is proposed for CES for Turkish is explained and several experiments are done. The first issue that must be faced is the complexity of Turkish. Therefore, the methodology starts with a pre-processing step in which each document is converted to UTF-8 format. The documents are parsed to its words by BoMorP and BoDis programs. Nouns of the documents are selected and represented in "bag-of-words" form. Then some clustering algorithms are applied to the bag-of-words. Since concepts can be defined by words, clustering similar words is considered to be useful for CES. The k-means algorithm with 100 clusters is determined as the best algorithm for this system. The clusters are assigned to the documents according to the words they contain. Then concepts are assigned to the clusters by human intervention. After that, documents are identified by the concepts via the clusters assigned to the documents.

After determining the methodology, experiments are applied and their evaluations are given. First of all, a corpus is selected and the methodology is applied to it. Then, testing strategies are determined that precision and recall method is used generally. Three types of testing are applied which are by words, by clusters, and by concepts. In the test by words, the words of the clusters which are assigned to the articles are compared with the nouns of the key files. As a result, the accuracy is 46 per cent. This is lower than expected. The assigned clusters of the articles are tested by the assigned clusters of the key files. 41

per cent accuracy is obtained in other words general topics of articles are defined with 41 per cent accuracy. This score is not quite high but higher than some similar studies. Lastly, after assigning concepts to the clusters, and determining the concepts of the articles via the clusters, they are tested by the concepts of the key files. As a result, 51 per cent of the assigned concepts are matched with the concepts of the key files whereas only 22 per cent of the assigned concepts are assigned correctly. The recall being too high may be due to this fact. Therefore, some other experiments are done by limiting the assigned concepts of the articles, but the results do not improve.

In this system, the first issue that must be faced is the complexity of Turkish which is an agglutinative language. The second issue is the abstractness of concepts. To the best of our knowledge, this study is the first concept extraction study for Turkish. This work can serve as a pioneering work in concept extraction field for agglutinative languages. The results are better than the studies related to this field.

As a future work, the methodology must be applied to a new corpus. Due to the fact that finding Turkish documents with their concepts or key phrases is not easy, and moreover creating such a comprehensive corpus takes too long time, the methodology is tested on only one corpus. By creating a new corpus, this study can be tested on it. In order to improve the methodology, other clustering algorithms, for example, supervised learning algorithms may be tried.

APPENDIX A: CLUSTERING

A.1. Document-Noun Matrix

The matrix is created from "bag-of-words" of *Gazi* corpus. Rows hold the number of the articles. Columns hold the words of the bag-of-words. The numbers in the table shows how many times the word in the column occurs in the article in the row. There are 1494 nouns in the bag-of-words, but we can show only eight of them in this table.

Table A.1. A sample from document-noun matrix from *Gazi* corpus

DocNo	**şekil**	**değer**	**sistem**	**sonuç**	**el**	**malzeme**	**işlem**	**ara**
Doc1	6	7	49	6	12	22	2	13
Doc2	18	19	21	15	7	0	38	1
Doc3	11	21	11	4	7	125	0	10
Doc4	27	3	0	7	13	18	19	6
Doc5	30	28	45	10	13	0	16	6
Doc6	39	10	0	7	1	9	1	6
Doc7	23	34	6	30	8	8	14	17
Doc8	27	39	3	24	21	2	0	6
Doc9	24	15	12	10	2	2	1	11
Doc10	25	49	0	30	16	0	0	7
Doc11	13	14	3	4	3	0	0	1
Doc12	12	6	0	1	2	0	0	6
Doc13	19	9	4	22	15	0	8	32
Doc14	31	42	109	20	19	0	0	0
Doc15	8	11	7	22	22	0	10	10
Doc16	6	39	2	30	19	39	16	14
Doc17	28	29	91	38	14	16	7	31
Doc18	31	10	0	15	0	87	6	8
Doc19	10	12	1	11	31	37	3	3
Doc20	5	22	20	18	17	1	31	12

Table A.1. A sample from document-noun matrix from *Gazi* corpus (contd.)

Doc21	3	6	5	8	16	32	1	2
Doc22	18	53	1	11	4	3	1	8
Doc23	6	18	5	6	2	3	0	3
Doc24	17	16	1	13	4	0	0	25
Doc25	27	11	4	47	26	17	27	12
Doc26	23	48	7	10	10	0	14	13
Doc27	6	10	57	9	12	0	8	4
Doc28	59	23	33	12	28	4	104	22
Doc29	18	21	0	4	8	1	0	18
Doc30	40	2	0	34	13	13	20	11
Doc31	10	5	0	3	3	6	12	6
Doc32	30	28	3	14	19	25	1	55
Doc33	20	11	5	6	6	0	0	8
Doc34	39	10	6	8	15	0	4	3
Doc35	12	17	0	12	5	10	0	2
Doc36	15	18	0	6	27	9	4	8
Doc37	56	73	122	14	55	0	6	11
Doc38	14	1	25	6	18	0	6	7
Doc39	33	19	0	12	10	25	6	7
Doc40	10	45	0	6	5	0	2	11
Doc41	26	5	4	4	5	0	7	8
Doc42	37	11	9	10	21	0	30	12
Doc43	9	5	8	9	2	5	3	22
Doc44	24	19	2	17	18	2	29	3
Doc45	7	4	0	4	2	0	0	6
Doc46	62	43	6	9	10	8	0	27
Doc47	19	15	40	24	23	0	0	10
Doc48	19	19	11	4	10	14	0	19
Doc49	16	58	28	28	22	1	14	3
Doc50	4	102	16	15	10	0	7	14
Doc51	26	21	67	8	2	68	8	15

Table A.1. A sample from document-noun matrix from *Gazi* corpus (contd.)

Doc52	35	10	76	13	11	2	88	16
Doc53	23	5	59	8	9	0	24	5
Doc54	15	7	20	2	4	2	0	2
Doc55	12	21	0	6	7	11	2	1
Doc56	21	9	9	2	4	19	13	5
Doc57	32	26	0	14	13	1	0	26
Doc58	21	7	6	10	22	38	9	18
Doc59	34	19	20	15	6	0	0	2
Doc60	23	20	6	3	6	0	33	2

A.2. Clusters and Words

The clusters of *Gazi* corpus in Table A.2 are created by Tanagra k-means algorithm. The best cluster number is determined as 100.

Table A.2. Cluster numbers and their words from *Gazi* corpus

cluster no	**words**
1	kriter, alternatif, stok, alım, yetenek, pratik
2	alan, uygulama, denetim, süreç, teknik, duvar, kabul, proje, bağlam, organizasyon, tesisat, panel, öneri, entegrasyon, perde, derz, kabuk, olgu, evre, bölme, birikim, site, geçirim, kategori, tünel, yaptırım, dokümantasyon, öz, çöp, halı, küf
3	uzman, konum, kütüphane, dahil, isim, tarif, tanıtım, bağlama, dikdörtgen, freze, yörünge, sakınca, sürü, taşlama, otomasyon, prizma, puma
4	negatif
5	numune, agrega, ağırlık, gazi, cilt, fak, mim, ocak, obruk, mühendis, saha, öze, klorür, alkali, şek, formasyon, kusur, dağ, don, emme, gri, kayaç, gnays, kuvarsit
6	karşı, kadın, çaba, para
7	inşa, gelecek, mahalle, bakan, sembol, çeşme, toplantı
8	ağ, karakter, sinir, karşılık, kod, harf, hedef, desen, yazı, beyin, türkçe, roman, ihmal, kabiliyet, ağı, taktir, font

Table A.2. Cluster numbers and their words from *Gazi* corpus (contd.)

9	kontrol, son, ünite, satır, komut, format, döngü, basamak, er, virgül, teker, mazak, telafi
10	yapı, gerek, dış, söz, oluşum, yâd, doküman, disiplin, delik, korunum, şap
11	kalite, boyut, nitelik, bileşen, hasar, kavram, ses, yalıtım, ilke, teşekkür, nem, konstrüksiyon, onarım, koordinasyon, içerik, baskın
12	model, referans, kazanç, birey, bit, küme, makale, un, suret, ima, topluluk, kümes, is, genetik, dümen, gemi, kargo, prosedür, gem, bellek, mutasyon
13	açı, kuvvet, talaş, sebep, uç, yaş, geometri, işleme, çap, kök, morfoloji, yarıçap, çözelti, düzlem
14	parametre, kâr, literatür, karınca, düğüm, şehir, tur, koloni, optimizasyon, arkadaş, iz, kez, bağıntı, yayın, yuva, meta, yiyecek, tüm, satıcı
15	yöntem, form, başarı, ikili, verim, çalışan, avantaj, seçenek, puan, gösterge, atölye, dezavantaj, hiyerarşi, yönetici, büro, ücret, beceri, gider, uygulanabilirlik, emir(ı), motivasyon, departman, parti, tahsis, terfi, indirim, kariyer, liderlik, ödül
16	iç, gün, zemin, cephe, özen, ilçe, tescil
17	kat, bina, blok, işlev, endüstri, fabrika, yerleşke, tekel, üniversite, depo, belediye, karadeniz, sigara, kolon, tütün, itibari, sanayi, miras, restorasyon, idare, inhisar, mira, müdürlük, gündem, kiremit, bodrum, hükümet, reji
18	hâl, resim, maske, başarım
19	yol, dakika, güvenlik, otel, merdiven, çizelge, önlem, koridor, yatak, asansör, lüks, örneklem, ölü, atıf, daire, tahliye, uyku, alarm, genişlik, konuk, süit
20	-
21	tespit, kimlik, teknoloji, fotoğraf, kablo, iletişim, kart, yüz, varsayım, sunucu, nüfus, hazne, vatandaşlık, kamera, öğrenci, vatandaş, istem, suç, dizüstü, organ
22	amaç, süre, temel, ürün, minimum, hak, aracılık, hazırlık, işleyiş
23	el, ifade, fonksiyon, biçim, sınır, denklem, sinyal, baş, tahmin, il, teorem, terim, tarz, denetleyici, et, pozisyon, türev, ispat, integral, bilim, diferansiyel, kısa, orantı, mertebe, lim, zarf, ihtiva, kontrolör, sunum, artı, öngörü
24	kayıt(ı), varlık, ayrıntı
25	problem, çözüm, hareket, aralık, başlangıç, liste, değişken, tabu, çöz, set, uzunluk, komşu, magazin, kombinasyon, koşu, tamsayı

Table A.2. Cluster numbers and their words from *Gazi* corpus (contd.)

26	taş, sıva, saat, alçı, yarı, dolgu, kalsiyum, anhidrit, fırın, harç, koruma, usta, bünye, doku, elek, fır, temmuz, jips, karkas, çelebi, defa, kristal, tehlike, hidrat, saf, hiza, kuşak, perdah, difraksiyon, ızgara, molekül, sorumlu, tekke
27	buhar, basın, termodinamik, net, dost, entropi, çıktı, yayılma, ozon, vade
28	hata, bölüm, yaklaşım, yarar, ad(ıı), belirti, baz, takip, yardımcı, yargı, metodoloji, arzu, iddia
29	tablo, standart, ilgi, üye, izin
30	durum, sayı, nokta, bölge, içeri, istasyon, şart, talep, mesafe, mevcut, aday, yakın, kısıt, senaryo, karakteristik, iklim, hizmet, kaza, imkân, gösteri, helikopter, tesis, ikmal, silah, deniz, kara, olay, teşkil, tim, lığ, müdahale, tank, intikal, sivil, teşkilat
31	yakıt, çevre, enerji, reaktör, karışım, kesit, demet, reaksiyon, nötron, rezonans, çubuk, eş, çekirdek, data, tesir, transport, tüp, termal, toryum, kor, uranyum, rezerv, atık, plütonyum
32	kent, tarih, kültür, kurum, kurgu, cami, ortaklık, çevri, medeniyet, yay
33	meydan, dönüşüm, park, rol, bulvar, öykü, hürriyet, başkent, geçmiş, engel, ideoloji, güven, havuz, kurul, heykel, kanıt, otopark, peyzaj, anı, imaj, prestij, yarışma, akşam, araba, sergi, taşımacı
34	aşama, tip, seçim, ekran, taşıt, bant, bugün, tipi, sayfa, tuş, mobilya, metre, faiz, satın, dal, uzaklık
35	insan, rahat, hafta, memnuniyet
36	faktör, ilgili, şikâyet
37	orta, merkez, türk, yüzyıl, faaliyet, pazar, balık, ticaret, gelenek, coğrafya, deneme, art, inanç, evrim, kale, vadi, köken, budist, hu, köy, tipoloji, asker, göçebe, hatun, kitabe, külliye, odak, ordu, kışlak, saray, sur(ıı), vergi, direk, göktürk, çağ, göl, islam, kaya, kule, türbe
38	şekil, algoritma, bura
39	kullanım, ilişki, ışık, ferah, müzik, çiçek, kalabalık, masa, alfa(ı), manzara, kişilik
40	bilgi, tasarım, taban, mekanizma, temsil, kural, unsur, saye, dil, şartname, torna, rapor, zekâ, çıkar, yetki, editör, taklit, gramer, katalog, modifikasyon
41	veri, ihtiyaç, altyapı
42	malzeme, uyum, dökme

Table A.2. Cluster numbers and their words from *Gazi* corpus (contd.)

43	esas, ilave, esna, balata, toz, fren, disk, metalurji, kurşun, oksit, kurt
44	kumaş, mukavemet, ip, pamuk, randıman, polyester, gabardin, giysi, iplik, korelasyon, dikim, merserize, materyal, gramaj, ilmek, kesik, uzam, atkı, çözgü, iğne, proses, etiket
45	maliyet, soru, istek, deyiş, yatırım, bilgin, operatör
46	zaman, iş, dikkat, an(ıı), tür, kutu, numara, öncelik, dizi, diyagram, çevrim, sütun, dilim, zincir, şey, yüklem, ham, hayal, dizin, ok, peş
47	yıllık, sofa, hacı, mutfak, etraf, kış, akıl, tavan, ahır, eğim, yaz, yazlık, güneş, soba, arsa, ders, kazan, saman
48	su, yöre, sülfat, dere
49	program, kalıp, yerleşim, plaka, şerit, aza, sac, zımba, çizim, dişi, uzantı, baskı, cıvata, diyalog, nalbant, delme, pim
50	âdet, köşe, birleşim, kavela, pencere, formül, rutubet, diş, kuru, regresyon, zıvana, not, kereste
51	hat, bilgisayar, yazılım, cihaz, bağlantı, kanal, donanım, seri, mesaj, arayüz, menü, dosya, servis, hafıza, erişim, trafo, kana, telefon, video, tampon, bacak, terminal
52	in, çimento, dünya, inşaat, sektör, kireç, santral, kül(ıı), uçucu, kömür, linyit, öte, kir, ekonomi, tasarruf, incelik, ikame, fil, kerpiç, tarım
53	gerçek, düzen, metin, kelime, felsefe
54	deney, dayanım, beton, metot, basınç, tahribat, eksen, örtü, hassasiyet, yazar, bileşik, şimşek, küp, çeki, sonda, kür, araştırmacı, çekiç, laboratuvar, mastar, silindir
55	taraf, hol, destek, hastane, klinik, kare, poliklinik, hasta, tahlil, ay, sağlık, yüzde, deneyim, kan, röntgen, kardiyoloji, algı, dermatoloji, kıyas, idrar, huzur, tedavi, muayene, sanat, fikir, bayan, bay, karanlık, stres, atmosfer, eser, verici, meslek, tabela, kasvet, sirkülasyon, teşhis, ziyaret, danışma, gösterim, sıkıntı, yorum, boğaz, buru, kulak, onay, psikoloji, ruh, sıklık
56	devlet, toplum, cadde, bahçe, gelişim, kânun, kuruluş, imar, aks, terk, amerikan
57	ara, yüzey, bağ, ısı, katsayı, transfer, pürüz, hesap, temas, düzenek, indis, simetri, şahin
58	yön, olanak, alışveriş

Table A.2. Cluster numbers and their words from *Gazi* corpus (contd.)

59	eğlence, dönem, tercih, gelir, cumhuriyet, etken, hayat(ı), tüketim, anlayış, sinema, ulus, iktidar, lokanta, politika, yaşantı, gazino, kamu, internet, kitle, radyo, rejim, balo, hâkimiyet, inisiyatif, çay, düğün, meyhane, baba, girişim, hegemonya, pavyon, ret
60	etki, hava, ortam, ömür, deformasyon, krater, regülatör
61	eksik, doğrultu, yönetim, tecrübe, mal, ekip, ayak, işletme, paha
62	araç, konfor, emniyet, kemer, katılım, koltuk, sağ, buton, cinsiyet, konsol, ayar, erkek, aksesuar, kul, skala, sol(ıı), tatmin, sürücü, takı, yolcu, otomobil, ölüm, uçak, yolculuk, dizayn
63	kapsam, boy, başlık, belge, bakır, iskân
64	sonuç, özellik, döküm, vakum, metal, alaşım, mekanik, yardım, gaz, sıcaklık, düşük, gözenek, netice, parçacık, gül
65	eleman, tekrar, akı, kompresör, sermaye, top, cin, valf
66	oran, birim, hacim, kum, briket, kak, nü, kül(ı), fiyat, civar, temin, aktivite, bağıl
67	mekân, plan, ölçek, canlı, can, kafe, televizyon, okul, yaya
68	neden, gürültü, ulaşım, ray, ölçüm, trafik, transit, gece, sefer, gündüz, harita, tren, alıcı, bariyer, lokomotif, otobüs, remel
69	yıl, doğu, güney, batı, kuzey
70	eğitim, katman, lazer, nöron, optik, kuyu, eşik, topla, ün, yâr
71	gerilim, akım, güç, şebeke, faz, yük, koşul, denge, özet, filtre, ideal, işletim, tel, simülasyon, dalga, anlık, kol, reaktif, şema, edim, teori, eşitlik, histerezis
72	kapı, ev, kanat, avlu, çoğunluk, görünüm, halk, kasa, sokak, süsleme, yıldırım, dışarı, göbek, kilit, zemberek, lamba, yaprak, çadır, çatkı, rölyef, tokmak, ayna, banyo, sol(ı), süs, bini, aksam, cihannüma, çıta, niş
73	üretim, seramik, tuğla, kompozisyon, silis, baca, ateş
74	sistem, yer, karar, yukarı, devam, ileri
75	seviye, vektör, koordinat, frekans, işaret, uzay, modülasyon, gen, indeks, köprü, motor, koordinatlar, stator, kartezyen, genlik, tatar, forma, izolasyon, platform, üçgen
76	vasıta, husus

Table A.2. Cluster numbers and their words from *Gazi* corpus (contd.)

77	örnek, düzey, çeşit, renk, ağaç, ton, değiş, etkileşim, yağ, cila, kestane, parafin, koruyucu, meşe, tik, akasya, armut, sedir, aritmetik, ışın, profil, sıvı, vernik, maksat, pigment, def, odun, ark, mum, radyasyon, absorbe, fiber, söğüt
78	sahip, oda, üst, dolap, yemek, balkon
79	sınıf, test, ülke, testi, tepki, katkı, birlik, döşeme, kenar, ek, çatı, yangı, alev, duman, direktif, ortak, komisyon, tabaka(ıı), levha, hariç, tanecik, boru, kılavuz, yürürlük, fire, gazete, kâğıt, lâmi, bakanlık, kütle, özelik, arduvaz, denk, dolaşım, inorganik, ekim, mayıs
80	potansiyel, kil, lif, mineral, tayin, rüzgâr, kalori
81	modül, değil, mantık, rakam, dâhi
82	değişim, sonra, aşağı, önce, madde, nehir, baraj, zarar, gözlem, debi, kesim, ask, ten, ağız, hidrolik, ağustos, pik, aylık, rezervuar, lük, askı, general, nisan
83	anlam, detay, oyun, mimar, tiyatro, karagöz, düşünce, görüş, sahne, sürek, güldürü, alay, hâkim, bilinç, duygu, seyirci, sözcük, varoluş, yabancı, yok, herkes, norm, strüktür, çelişki, dekor, giyim, gölge, hacivat, kavuk, kırcı, metafor, pişekâr, yaratı
84	işlem, parça, tezgâh, sıra, imalat, robot, hücre, kısım(ıı), ad(ı), operasyon, tabla, kütük, nesne, imal, sur(ı), piyasa, alacak, ekipman, yurt, darboğaz, tab
85	analiz, tanım, çerçeve, eğri, kapasite, dinamik, limit, deprem, as, risk, atım, sönüm, olasılık, cins, ihtimal, medya, açıklık, kiriş, şiddet, adet
86	performans, göz, ön, gereksinim, beklenti, dolay, aktarım
87	yangın, konu, yönetmelik, mevzuat, cam, saniye, hüküm
88	alt, grup, dağılım, bakım, grafik, çizgi, istatistik, çeyrek, ortanca, sin(ıı)
89	yan, adım, arıza, direnç, toprak, transformatör, benzetim, sarı, elektrik, sargı, diren, darbe, sarım, kaçak, benzeşim, manyetik
90	fark, yaşam, ana, konut, kişi, aile, bulgu, misafir, eylem, donatı, salon, mülkiyet, arz, eşya, apartman, kitap, enstitü, hipotez, milyar, dekorasyon, fakülte
91	ölçü, tolerans, montaj, moment, tol, prensip, sentez, monte, varyasyon, halka, dize, müsaade
92	devre, anahtar, sıfır, iletim, periyot, kondansatör, bobin, indüksiyon, dönüş, ısıtıcı
93	yapım, doğa, yağmur, pano

Table A.2. Cluster numbers and their words from *Gazi* corpus (contd.)

94	kaynak, tane, makine, bar, perlit, çelik, marka, alın, girdi, sem(ı), mikroskop, elektron, karbon, çene, enjeksiyon
95	önem, sorun, ortalama, derece, şirket, anket, personel, firma, cevap, paket, danışman, fayda, sapma, dağıtım, hayır, sap, uyarlama, ciro, yanlış, pay, müşteri, finans, muhasebe, ambar, rekabet, sipariş, tedarik, eleştiri, strateji, adaptasyon, görüşme, işletmen
96	hız, takım, karbür, plastik, kaplama, dinamometre, nikel, alet, altın, kist, testere, asıl, tarama, tavsiye, yanak
97	boya, estetik, arka, üzeri, görü, tuvalet
98	eğilim, pas, kâp, dev, yığıntı
99	değer, toplam, matris, ölçüt, görev, kademe, işçi, alıç, ar
100	miktar, boşluk, kısım(ı), görüntü, bileşim, at, besleme, pres, element, atom, çekme

APPENDIX B: DOCUMENTS AND CLUSTERS

B.1. Articles and Assigned Clusters

The clusters are assigned to the articles in *Gazi* corpus according to nouns they contain. Table B.1 shows the article numbers and their assigned cluster numbers.

Table B.1. Article numbers and their assigned clusters from *Gazi* corpus

doc_no	**clusters no**
1	93, 36, 11, 2, 10
2	18
3	80, 63, 74, 22, 29, 79, 87, 42, 10
4	13
5	4, 76, 74, 38, 12
6	24, 97, 78, 16, 93, 72
7	54
8	85
9	89
10	76, 30
11	4, 92
12	7, 69, 32, 33
13	6, 36, 35, 62
14	38, 68
15	38, 70
16	4, 77
17	38, 22, 61, 28, 45, 81, 41, 95
18	43
19	73, 80, 52
20	38, 25
21	74, 86, 11
22	4, 88, 50
23	19, 87
24	36, 39, 78, 90
25	64
26	38, 27
27	65
28	38, 46, 74, 84
29	82
30	4, 100
31	60
32	42, 57
33	69, 16, 17
34	38, 75
35	66
36	96
37	76, 4, 38, 28, 23, 74
38	21
39	94
40	4, 44
41	76, 24, 51
42	76, 74, 38, 8
43	53, 83
44	31
45	6, 58, 56, 67, 59
46	36, 35, 4, 58, 67, 39, 55
47	38, 14
48	69, 47
49	76, 91
50	58, 1, 15, 99
51	4, 1, 34, 45
52	41, 38, 74, 3, 84
53	40
54	24, 53, 63, 32, 37
55	98
56	49
57	4, 88, 5, 48
58	48, 16, 26
59	4, 71
60	9

B.2. Key Files and Clusters

The clusters are assigned to the key files of the articles in *Gazi* corpus according to nouns they contain. Table B.2 shows the key file numbers and their assigned cluster numbers.

Table B.2. Key file numbers and their assigned clusters from *Gazi* corpus

key file no	**clusters no**
1	88, 74, 86, 1
2	74
3	10, 42, 79, 87
4	13, 64
5	81, 2, 40, 12, 38
6	37, 72, 78
7	54
8	85, 10
9	89, 16, 92, 51, 12
10	74, 34, 25
11	92, 9
12	33, 67, 59
13	33, 62
14	68, 74
15	88, 36, 8, 70
16	42, 77
17	94, 95, 81
18	42, 43
19	52, 79, 42
20	84, 74, 34, 25
21	11, 10, 28, 86, 1
22	72, 50
23	87, 19
24	90, 67, 16, 40, 65
25	64, 13, 66, 49
26	8, 27, 74, 64
27	57
28	84, 3, 74
29	60, 17, 82, 71, 19, 77, 92, 85
30	64, 13, 66, 49, 100
31	64, 13, 96, 60
32	57, 58, 9, 90, 54
33	17, 30
34	75, 38
35	54, 66
36	96, 13, 57
37	2
38	8, 19, 74
39	94, 71, 46, 64
40	44, 11
41	23, 51, 0
42	8, 100, 13
43	83, 11
44	31
45	59, 2, 67
46	55, 86
47	14, 25
48	72, 62, 85, 17, 86
49	91, 85
50	99, 15, 44
51	74, 55, 42, 1
52	84, 3, 74, 40
53	40, 3, 74, 34
54	37, 32, 12
55	13, 96, 57
56	49, 40
57	79
58	26, 37, 72
59	71
60	84, 3, 41, 51, 55, 40, 19

APPENDIX C: DOCUMENTS AND CONCEPTS

C.1. Articles and Concepts

After assigning concepts to the key files and to the clusters by a human specialist, the concepts are assigned to the articles according to their assigned clusters. Table C.1 shows the article numbers, the concepts assigned to them and number of repeated concepts.

Table C.1. Articles, their concepts and concept repetition count from *Gazi* corpus

doc no	**concept**	**count**
1	yapı	3
1	malzeme	2
1	performans	2
1	yer bilim	1
2	performans	1
3	yapı	1
3	sistem	2
3	performans	1
3	malzeme	4
3	ev	2
3	yer	1
3	yasa	2
3	yer bilim	1
3	yangın	1
4	malzeme	1
4	geometri	1
5	model	1
5	algoritma	2
5	sistem	1
5	yer	1
5	araç	1
6	ev	3
6	sistem	1
6	yapı	3
6	yer bilim	1
6	malzeme	1
7	beton	1
7	malzeme	1
8	yer bilim	1
8	analiz	1
9	elektrik	1
10	askeri	1
10	benzetim	1
10	araç	1
11	iletim	1
11	elektrik	1
12	yapı	3
12	inşaat	2
12	medeniyet	1
12	mekan	1
12	şehir	1
12	ulaşım	1
13	insan	1
13	performans	1
13	güç iktidar	1
13	ulaşım	1
13	araç	1
14	algoritma	1
14	ulaşım	1
15	algoritma	1
15	sinir ağı	1
16	ev	1
16	inşaat	1
17	sistem	3
17	model	1
17	algoritma	1
17	bilgisayar	1
17	finans	3
17	mantık	1
17	analiz	1
18	malzeme	1

Table C.1. Articles, their concepts and concept repetition count from *Gazi* corpus (contd.)

18	araç	1
19	malzeme	3
19	inşaat	2
19	yer bilim	2
20	model	1
20	matematik	1
20	algoritma	1
21	malzeme	1
21	performans	2
21	sistem	1
21	yer	1
22	yapı	1
22	inşaat	1
22	matematik	1
23	ev	1
23	ulaşım	1
23	yangın	1
23	yasa	1
24	performans	1
24	ev	3
25	ısı	1
25	malzeme	1
26	ısı	1
26	enerji	1
26	algoritma	1
27	iletim	1
28	algoritma	1
28	yapı	1
28	zamanlama	1
28	sistem	1
28	yer	1
28	malzeme	1
29	su	1
30	malzeme	1
31	malzeme	1
31	yer bilim	1
32	malzeme	1
32	ısı	1
33	ev	2
33	inşaat	1
33	yapı	1
33	ulaşım	1
34	algoritma	1
34	elektrik	1
34	iletim	1
35	malzeme	1
35	matematik	1
36	malzeme	1
36	yapı	1
37	sistem	3
37	matematik	1
37	model	1
37	algoritma	1
37	yer	1
37	araç	1
38	model	1
38	biyometrik	1
39	elektrik	1
39	insan	1
40	malzeme	1
40	tekstil	1
41	sistem	1
41	iletim	1
41	bilgisayar	1
41	araç	1
42	algoritma	1
42	sistem	1
42	yer	1
42	araç	1
42	sinir ağı	1
42	yazı	1
43	yazı	2
43	eğlence	1
44	ısı	1
44	iletim	1
44	nükleer enerji	1
45	yapı	1
45	yasa	1
45	finans	1
45	eğlence	2
45	güç iktidar	1
45	ev	1
46	insan	1
46	performans	1
46	ev	2
46	yapı	1
46	inşaat	1
46	hastane	1
46	finans	1
46	eğlence	1
47	performans	1
47	optimizasyon	1

Table C.1. Articles, their concepts and concept repetition count from *Gazi* corpus (contd.)

47	algoritma	1
48	ev	1
48	inşaat	1
48	ulaşım	1
49	araç	1
49	performans	1
50	performans	2
50	güç iktidar	1
50	finans	1
50	analiz	1
50	matematik	1
51	performans	1
51	ulaşım	1
51	ev	1
51	bilgisayar	1
51	finans	1
52	malzeme	2
52	algoritma	1
52	sistem	2
52	yer	1
53	algoritma	1
53	iletişim	1
54	sistem	1
54	yapı	1
54	inşaat	1
54	medeniyet	1
54	ev	2
54	türk	1
54	tarih	1
54	şehir	1
54	yazı	1
54	malzeme	1
55	yer bilim	1
55	malzeme	1
56	yapı	1
56	inşaat	1
57	su	1
57	yer bilim	2
57	malzeme	1
57	yer	1
57	matematik	1
58	ev	1
58	yapı	1
58	malzeme	1
58	su	1
58	yer bilim	1
59	elektrik	1
59	iletim	1
60	bilgisayar	1

C.2. Key files and Concepts

Some concepts are assigned to the key files according to words they contain by a human specialist. Table C.2 shows the key file numbers and the concepts assigned to them.

Table C.2. Key file numbers and their concepts from *Gazi* corpus

key file no	**concepts**
1	sistem, performans
2	sistem
3	yapı, malzeme
4	yapı
5	mantık, model
6	ev, türk
7	beton, matematik
8	yapı, analiz
9	iletim
10	yer
11	elektrik
12	şehir, ideoloji
13	ulaşım
14	ulaşım
15	sinir ağları, performans
16	malzeme
17	kaynak, mantık
18	malzeme
19	malzeme
20	sistem
21	performans, yapı
22	yapı
23	yangın
24	yapı
25	malzeme
26	sinir ağları
27	malzeme
28	sistem
29	nehir, elektrik
30	malzeme
31	yapı
32	ısı
33	yapı
34	mantık, iletim
35	beton, yapı
36	yapı
37	teknik
38	sistem
39	malzeme
40	performans
41	iletim
42	sinir ağları
43	eğlence
44	ısı
45	yapı, eğlence
46	yapı, performans
47	optimizasyon
48	ev, performans, analiz
49	analiz
50	sistem
51	sistem, malzeme
52	sistem, zeka
53	sistem
54	yapı, şehir
55	yapı, beton
56	inşaat
57	malzeme
58	ev, türk
59	elektrik
60	bilgisayar, ulaşım

REFERENCES

1. Mengüşoğlu, T., *Felsefeye Giriş,* 7th Edition, Remzi Kitabevi, Istanbul, 1992.

2. Crangle, C., Zbyslaw, A., Cherry, M. and Hong, E. L., "Concept Extraction and Synonymy Management for Biomedical Information Retrieval", *Proceedings of the Thirteenth Text REtrievel Conference (TREC 2004),* 2004.

3. SPSS Inc., "Mastering new challenges in text analytics", *SPSS Technical Report*, MCTWP-0109, 2009.

4. Gelfand, B., Wulfekuhler, M. and Punch W.F. III., "Automated Concept Extraction from Plain Text", *Papers from the AAAI 1998 Workshop on Text Categorization*, 1998.

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

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

7. Villalon, J. and Calvo, R.A., "Concept Extraction from Student Essays, Towards Concept Map Mining", *ICALT 2009,* pp. 221-225, 2009.

8. Navigli, R. and Velardi, P., "Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites", *Computational Linguistics,* Vol. 30, pp. 151-179, 2004.

9. Bourigault, D. and Jacquemin, C., "Term extraction + term clustering: An integrated platform for computer-aided terminology", *EACL*, 1999.

10. Bichindaritz, I. and Akkineni, S., "Concept Mining for Indexing Medical Literature", *Lecture Notes in Computer Science*, Vol. 3587, pp. 682-692, 2005.

11. Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C. and Nevill-Manning, C. G., "KEA: Practical Automatic Keyphrase Extraction", *Working Paper 00/05*, Hamilton, New Zealand: University of Waikato, 2000.

12. SPSS Inc., "Gaining Full Value from SPSS Text Analysis for Surveys", *SPSS Technical Report*, GVSTWP-1008, 2008.

13. Moens, M.F. and Angheluta, R., "Concept Extraction from Legal Cases: The Use of a Statistic of Coincidence", *Proceedings of the Ninth International Conference of Artificial Intelligence and Law*, 24-28 June, 2003.

14. Bing, J., "Performance of Legal Text Retrieval Systems: The Curse of Boole", *Law Library Journal*, vol. 79, pp.187-202, 1987.

15. Rissland, E. L., Skalak, D. B., and Friedman, M.T., "Bankxx: Supporting legal arguments through heuristic retrieval", *Artificial Intelligence and Law*, 4 (1):1-71, 1996.

16. Winkels, R., Bosscher, D., Boer, A. and Hoekstra, R., "Extended conceptual retrieval", *Legal Knowledge and Information Systems: Jurix 2000: The Thirteenth Annual Conference*, pp. 85-97, IOS Press, Amsterdam, 2000.

17. Wang, J., Peng, H. and Hu, J., "Automatic Keyphrase Extraction from Document Using Neural Network", *Advances in Machine Learning and Cybernetics*, Springer, Berlin, Heidelberg, pp. 633-641, 2006.

18. Turney, P., "Learning to Extract Keyphrases from Text", *National Research of Council of Canada*, 1999.

19. Quinlan, J.R., *C4.5: Programs for machine learning*, California: Morgan Kaufmann, 1993.

20. Lovins, J.B., “Development of a Stemming Algorithm”, *Mechanical Translation and Computational Linguistics*, vol. 11, pp. 22-31. 1968.

21. Porter, M.F., “An Algorithm for Suffix Stripping”, *Program; Automated Library and Information Systems*, vol. 14 (3), pp. 130-137, 1980.

22. Whitley, D., “The GENITOR Algorithm and Selective Pressure”, *Proceedings of the Third International Conference on Genetic Algorithms (ICGA-89)*, pp. 116-121, California: Morgan Kaufmann, 1989.

23. Rohini, U. and Ambati, V., “Extracting Keyphrases from Books Using Language Modeling Approaches”, *Proceedings of the 3rd ICUDL*, Pittsburgh, USA, 2007.

24. Tomokiyo, T. and Hurst M., “A Language Modeling Approach to Keyphrase Extraction”, *Proceedings of the ACL 2003 Workshop on Multiword Expressions*, pp. 33-40, Morristown, NJ, USA, 2003.

25. Provalis Research, *WordStat v6.0 Content Analysis and Text Mining Help File*, Montreal, Canada, 2009.

26. Yergeau, F., *UTF-8, A Transformation Format of ISO 10646*, Network Working Group, November, 2003.

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

28. Sak, H., Güngör, T. and Saraçlar, M., “Morphological disambiguation of Turkish text with perceptron algorithm”, *CICLing 2007*, vol. LNCS 4394, pp. 107-118, 2007.

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

30. Ozgur, A., *Supervised and Unsupervised Machine Learning Techniques for Text Document Categorization*, M.S. Thesis, Boğaziçi University, 2004.

31. Borgatti, S.P., *How to Explain Hierarchical Clustering,* University of South Caroline, http://www.analytictech.com/networks/hiclus.htm, 1994.

32. Rakotomalala, R., "TANAGRA: A Free Software for Research and Academic Purposes", *Proceedings of EGC 2005*, RNTI-E-3, vol. 2, pp.697-702, 2005.

33. *Journal of The Faculty of Engineering and Architecture of Gazi University,* Vol. 21 Nr. 1, Nr. 2, Nr. 3, Nr.4 and Vol. 20 Nr. 1, Nr.2, Nr.3, 2006.