A TURKISH NATURAL LANGUAGE INTERFACE FOR THE SEMANTIC WEB: A CASE STUDY ON TURKISH UNIVERSITIES

A MASTER'S THESIS

in

Computer Engineering

Atilim University

by

EKREM ÇAĞLAR YILMAZ

AUGUST 2008

A TURKISH NATURAL LANGUAGE INTERFACE FOR THE SEMANTIC WEB: A CASE STUDY ON TURKISH UNIVERSITIES

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

ATILIM UNIVERSITY

BY

EKREM ÇAĞLAR YILMAZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF COMPUTER ENGINEERING

AUGUST 2008

ABSTRACT

A TURKISH NATURAL LANGUAGE INTERFACE FOR THE SEMANTIC WEB: A CASE STUDY ON TURKISH UNIVERSITIES

Yılmaz, Ekrem Çağlar

M.S., Computer Engineering Department

Supervisor: Asst. Prof. Dr. Çiğdem Turhan

August 2008, 42 pages

When a user needs to reach desired information from the Internet, millions of documents with unstructured information that exist on the web cause problems. One of the current trends with the Internet is the Semantic Web technology which stores information in a machine-readable format since information is stored along with the knowledge about the information. The purpose of the Semantic Web technology is to make it easier to reach the information for the users that use the search engines. As the Semantic Web technology gains popularity, Natural Language Processing Interfaces became popular once again because they increase the expressive power of users for information retrieval since users are able to ask the questions in their native language.

In this thesis, a prototype of a Turkish Natural Language Interface for the Semantic Web is developed and the problems encountered in the creation of such a system are expressed while suggesting possible solutions. The domain of the developed system is the Turkish Universities with related information for Turkish high school students and university candidates who are novice users with little knowledge about computer technologies. The system is tested on users and the results are evaluated.

Keywords: Semantic Web, Natural Language Processing

ÖZ

ANLAMSAL AĞ İÇİN TÜRKÇE DOĞAL DİL İŞLEME ARAYÜZÜ: TÜRK ÜNİVERSİTELERİ ÜZERİNE ÖRNEK ÇALIŞMA

Yılmaz, Ekrem Çağlar

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Prof. Dr. Çiğdem Turhan

Ağustos 2008, 42 sayfa

Kullanıcı internet aracılığı ile hedeflediği bilgiye ulaşmak istediği zaman, ağ’da bulunan hiçbir yapıya uymayan milyonlarca farklı bilgi problem yaratmaktadır. İnternetteki mevcut akımlardan biri olan Anlamsal Ağ teknolojisi, bilginin yanı sıra bilgi hakkında bilgiyi içermesi sebebiyle, bilgileri makine-okunabilir formatta saklamaktadır. Anlamsal Ağ teknolojisinin amacı, bilgiye arama motoru vasıtasıyla ulaşmak isteyen kullanıcılara kolaylık sağlamaktır. Anlamsal Ağ teknolojisi popülerlik kazandıkça, Doğal Dil İşleme Arayüzleri yeniden popüler olmaya başlamıştır. Çünkü Doğal Dil İşleme Arayüzleri ile kullanıcılar sorularını anadillerinde sorarak bilgi araştırması yaparken kendilerini daha iyi ifade edebilmektedirler.

Bu tez çalışması kapsamında, bir Anlamsal Ağ için Doğal Dil Arayüzü prototipi geliştirilmiş ve sistem oluşturulurken karşılaşılan sorunların belirlenmesi ve bu sorunlar için muhtemel çözüm önerilerinin oluşturulması amaçlanmıştır. Geliştirilen sistemin içeriğinde Türk Üniversiteleri hakkında bilgiler bulunmaktadır ve bu bilgiler bilgisayar okur-yazarlığı başlangıç seviyesinde olan lise öğrencileri ve üniversite adayları hedef alınarak seçilmiştir. Sistem, kullanıcılar üzerinde test edilmiş ve sonuçlar değerlendirilmiştir.

Anahtar Kelimeler: Anlamsal Ağ, Doğal Dil İşleme

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Nowadays computer usage is necessary in every area of life, and the Internet allows us to generate, exchange, share and manipulate information in an uncountable number of ways on a global basis.

The Internet has millions of documents including e-books, papers, music, advertisements etc., and this uncountable number of documents causes numerous problems such as not being able to find required information among a large number of irrelevant web pages. For such problems encountered in searching information, web search engines became popular to find specific information.

The search engines' working principle is storing information about many web pages, where the contents of each page are analyzed to determine how it should be indexed. The usefulness of a search engine depends on the relevance of the result set it returns. While there may be millions of Web pages that include a particular word or phrase, some pages may be more relevant than others. The search engine cannot realistically decide which web page is more relevant for a given subject. In addition, among the results retrieved by the search engine, there are numerous outdated and repeated documents. The drawbacks of the search engine have led to a new solution in correct and easy information retrieval called the Semantic Web [1].

The Semantic Web is a new approach to store the data in machine-interpretable manner [2]. It stores information and the knowledge semantically i.e. relationships between data objects with the help of its syntax. To store the data objects and their relationships, Semantic Web needs an ontology that defines data models in terms of classes, subclasses, and properties. Semantic Web provides a more declarative, flexible and powerful information retrieval mechanism.

For a Semantic Web to be used effectively, a user-friendly interface is needed so that even a user with little computer know-how can find what they looking for. To develop such a user-friendly interface Natural Language Processing (NLP) techniques can be utilized.

NLP is an established subfield of artificial intelligence and computer science concentrated on analysis, understanding and parsing of a natural language. With the help of NLP, the communication power increases between humans and computer systems, and usage of such systems becomes easier especially for novice users since they use their native language [3]. There are two main fields of NLP; natural language generation and understanding systems. Natural-language-generation systems convert information from computer database into human language, whereas natural-language-understanding systems receive a sentence from human language and convert it into formal representations that are understandable by computer programs [4] [5].

In this thesis, a Turkish natural language interface for an academic Semantic Web is developed. Using this interface, a Turkish student who is about to set their preferences in choosing a university after taking the university exam, can find correct and relevant information about Turkish universities by asking questions in the Turkish language.

In the following two chapters, the literature background of this thesis is detailed. Natural Language Processing overview and problems that arise in processing the Turkish language is presented in Chapter 2 and the process of creation of the Semantic Web is summarized in Chapter 3. In Chapter 4, the design of the system is presented, followed by Chapter 5 which details its implementation. Finally, the Conclusion and the future work proposed for this thesis is given in Chapter 6.

2. NATURAL LANGUAGE PROCESSING

2.1 Overview

NLP is concentrated on increasing the communication between humans and computers. NLP is an important technology which may cause changes in the behavior of human-computer interaction [3] [6]. With the help of NLP, computer technologies can participate in different areas of life.

Some research and development areas of NLP in computer technologies can be listed below;

- Natural Language Understanding
- Natural Language Generation
- Speech or Voice recognition
- Machine Translation
- Spelling Correction and Grammar Checking

Reflections of these research areas on to the computer technologies can be sampled as follows;

- Finding mistakes of the words or sentences in a document written by a user with word-processors using autocorrect support,
- Querying databases in natural language instead of using a programming language such as SQL,
- Spontaneous language translation between users who speak different languages.

To reach these goals in computer technology with NLP, some knowledge about the processed language is needed [4] [6]:

- *Phonetic and Phonological knowledge:* realization of words as sounds.
- *Morphological knowledge:* the construction of words from basic meaning units called phonemes.
- *Syntactic knowledge:* the word orders to form sentences.

- *Semantic knowledge* which deals with what a word means and the combination of these meanings in sentences to form the meaning of sentences.
- *Pragmatic knowledge* which deals with the usage of sentences in different contexts and interpretation of sentences according to context.

2.2 Turkish Language

There have been numerous NLP research on different languages, yet the amount of research on the Turkish language is still few. In fact, Turkish which belongs to the Altaic branch of the Ural-Altaic family is one of the most widely spoken languages in the world, distributed over a large area in Europe and Asia.

In the Turkish language, the word order is S (subject) + O (object) + V (verb). Turkish is a free constituent order language, and the word order can be grouped as follows;

- *Ordered sentence*
    - Verb is located at the end of the sentence
    - Subject (Özne)+Object (Nesne)+Verb (Yüklem)
    - Example: “Ali okula gitti”

- *Unordered sentence*
    - Verb is not located at the end of the sentence
    - Verb (Yüklem) +Subject (Özne)+Object (Nesne)
    - Example: “Gitti Ali okula”

“Sentences with different word order reflect different pragmatic conditions, in that, topic, focus and background information conveyed by such sentences differ” [7].

Due to word order changes, information retrieval in Turkish is harder and more stressful when compared to English.

In Turkish, there are two group of suffixes defined as follows;

- *Constructive suffixes (yapım ekleri)*
    - forms a new dictionary-word from an old one
    - -mek, mak, -van
    - Example: yap-mak, yay-van
- *Inflectional suffixes (çekim ekleri)*
    - allows a dictionary-word to take its proper place in a sentence
    - -ca, -ce, -ça, -çe
    - Example: Kısaca, kolayca, sadece

NLP has been applied to the Turkish language in different applications, such as natural language understanding, machine translation, etc., some of which will be listed in the following section.

2.3 Previously Developed Natural Language Processing Systems

SHRDLU (1971): a system for computer understanding of English. The system answers questions, executes commands, and accepts information in normal English dialog [8].

ELIZA (1966): a computer program which studies natural language communication. It translates the answer of a person into question form and generates communication in the role of a "Rogerian psychoanalyst" [4].

LUNAR (1998): a software prototype developed by NASA. Lunar is used by geologists who are allowed to ask questions about chemical compositions of lunar samples [4].

TOY (2001): an open source Prolog-based software platform which provides a base for new applications involving the processing of Turkish language [9].

TUJA (2004): is a natural language interface for the generation of Java Class Skeleton. It utilizes TOY's morphological analyzer in the interface [5].

TUSA (2003): is a natural language interface used for an online personal calendar. It involves the morphological analyzer/generator of TOY with some modifications [10].

NALAN-TS (2003): is a Turkish natural language query interface that is used for SQL databases. NALAN-TS uses Turkish and database dictionaries in a TOY-based application [11].

Hakia (2007): is a general purpose Semantic Search engine via Natural Language Interface for English and Turkish [12].

PANTO (2007): stands for Portable nAtural laNguage inTerface to Ontologies, which provides users about acquiring the needed information from formally defined ontologies [13].

Ballhysa (2000): is a natural language processing infrastructure for Turkish based on TOY to produce a prototypical sentence-level translator between English, Albanian and Turkish [6].

Bilsel (2000): is developed as a “poem expert” for analyzing Turkish folk poems in terms of their rhyme and meter properties [6].

3. SEMANTIC WEB

3.1 Overview

The development of World Wide Web, have provided us with a huge amount of data. Managing and understanding this data is not a trivial task when we consider all the available data sources. To illustrate this, the workers in different sectors are sending and receiving hundreds of e-mails in each day. Dealing with this huge amount of data consumes most of the time of the people in business. Therefore, efficient tools and techniques are required to be able to deal with this data. Additionally, integration is an important challenge for IT managers. Integration is important to achieve a common view of all the data and understand their relationships [16]. Reaching data through the Internet is time-consuming as well.

For the problems encountered in searching information, Web search engines became popular to find specific information. In general search engines keep indexed data about many of the Web pages according to the page contents.

With search engines, the indexed data about web pages are stored in an index database for use in later queries. When a user enters a query into a search engine (typically by using keywords), the engine examines its index and provides a listing of best-matching web pages according to its criteria, usually with a short summary containing the document's title and sometimes parts of the text. Most search engines support the use of the Boolean operators AND, OR and NOT to further specify the search query.

One of the most important features lacking in the current web technologies is a link between the texts and external lexical resources encoding the semantic information. Some of the common problems in keyword-based information retrieval systems can be defined as follows;

- *Searching information:* Due to term ambiguity, the existing keyword-based search, retrieves irrelevant information and also misses some of the information due to storing similar concepts related to the material under different terms. For example, when we search the keyword ‘Jaguar’ in keyword-based search we will see a linear list of results which contains ‘Jaguar’ key in somewhere. This list will include the big-cat information and Jaguar car related information. In order to get the required data we have to manage the data externally.

- *Extracting information:* With current technologies, people have to read and browse the textual or other kinds of representations extensively to extract relevant information. Software agents do not effectively assist with this type of information retrieval and also they fail to integrate information from different sources.

- *Maintenance* of large storage of weakly-structured text is difficult and time-consuming. According to an academic study, the volume of information in the public Web tripled in 2003 when we compare to the volume of information in 2000 [16] [17].

Adaptation of information storage (i.e. websites) which has dynamic reconfiguration of data due to changing user profiles is not yet fully mastered.

Nowadays, with the current trends in computer technologies, the search engines started to focus on new information retrieval approaches which provide correct and easy access to reach the requested data, and provide more relevant data result sets. This new technology is called the Semantic Web.

Semantic Web provides the integration and combination of data among various kinds of data resources for the Web. The Web was initially designed for human processing, but the following trends in technologies and the needs for the increase in data-processing power, led the Web content to transfer itself into machine-readable format encoding semantic information [18].

The Semantic Web technology is the product of this thought, developed by Timothy John Berners-Lee. The Semantic Web tools have been recommended by the World Wide Web Consortium since 2004 [19] [20] [21].

Semantic Web applications utilize ontologies. A common definition of ontology is: "An ontology is an explicit and formal specification of a conceptualization of a domain of interest" [22]. Ontologies provide the skeleton for the Semantic Web and also for the formalization of the knowledge. Semantic Web with the help of ontologies provides machine-processable semantics of data and information resources that can be used between different agents which are software and people.

Since the data on the web has machine-processable format, it is almost considered as the reinvention of the World Wide Web's infrastructure on at least the scale of the original web [21] [23].

3.2 Ontology

Ontologies are the essentials of the Semantic web and can be accepted as metadata since they include explicitly represented semantics of data for the machine-processable data approach. Ontology provides the basis for information extraction and reasoning. Ontologies help people and computers access the required information needed by including explicit links between the form and the content of the information. By the help of this content-based data model, achievement of access, interoperability, and the communication across the different platforms in the web is provided [18] [23] [24].

The ontology concept is getting widely used in computer science and the process of its development is often a cooperative process [24] [25] [26] [27]. Ontology technology has some similarities with databases and the object-oriented paradigm. The difference between an ontology and a database is that an ontology does not store data but gives a map for the stored data. In a database, the data is stored in the database and the user receives exact answer matching with the user query. But ontology does not return the exact information since it has a reasoning process and it generates the data to be returned according to the ontology representation [27] [28].

Ontologies also have classes and inheritance principles similar to the object-oriented paradigm. But in ontologies multiple-inheritance is not possible [29].

Experts in ontology engineering and domain of interest are required to create and manage ontologies. Some sort of semiautomatic approach to ontology construction is through the use of knowledge discovery techniques. In generation of ontologies, metadata extraction is a crucial and time-consuming process. Extracting metadata from textual data and using this data to annotate the text is possible by the use of information extraction techniques based on natural language analysis. The evolution of ontologies is of key importance, because ontologies should change with the changes in the knowledge and usage of the system [16].

To illustrate and describe the ontology concept more clearly we can give an ontology example from medical ontologies [30]. UMLS (Unified Medical Language System) is a large database designed to integrate some biomedical terms collected from various sources. Metathesaurus, Semantic Network and Specialist Lexicon are three parts of UMLS. The Metathesaurus includes biomedical information about each of the terms in UMLS. The definition, source and some semantic properties such as concept synonym, siblings and parents, or relationships between terms are specified in Metathesaurus part of UMLS. The Specialist Lexicon contains syntactic information about biomedical terms. The Semantic Network is a top-level ontology. This ontology describes biomedical concepts and relationships among these concepts.

3.3 Tools used in Semantic Web Technology

A definition of an ontology is as a 4-tuple (C, R, I, A):

- 'C' is a set of concepts,
- 'R' a set of relations,
- 'I' a set of instances,
- 'A' a set of axioms [16] [31].

Ontologies can be developed using special languages and tools described in the following sections.

3.3.1 XML

XML refers to the extensible mark-up language [1]. It is a meta-language which can be used for defining the mark-up languages which are specific for certain applications. It was basically designed to transport and store data on the Web and elsewhere needed. It stores the data in plain text format, and this provides software- and hardware-independent data storage. With XML, a user can define his/her own tags, and also XML provides a data format to define structures in a document. XML Schema is used to specify a vocabulary to define new structures in an XML document by combining tags [1] [20] [32] [33].

The following Internet languages are defined with XML [1];

- XHTML the latest version of HTML (Hypertext mark-up language)
- WSDL is used for describing available web services
- WAP and WML used as markup languages for handheld devices
- RSS languages for news feeds

- RDF and OWL for describing resources and ontology
- SMIL for describing multimedia for the web

Each XML document is made up of an ordered and labeled structure, and with XML, all kinds of data structures can be encoded explicitly with the syntax. XML does not keep information about usage and the meaning of the data [1] [20].

XML has a very flexible text format which is derived from SGML (ISO 8879), and is a W3C Recommendation since Feb. 10, 1998 [19].

3.3.2 XPath

XPath is a language for finding information in an XML document, and XPath is used as syntax for defining parts of an XML document. For this purpose, XPath uses path expressions to navigate in XML documents and XPath is a W3C standard [1].

3.3.3 XQuery

XQuery is a query language used for XML documents like SQL for databases. XQuery is built on XPath expressions and supported by all the major database engines (IBM, Oracle, Microsoft, etc.). XQuery is a W3C Recommendation [1].

3.3.4 RDF and RDF Schema

RDF refers to Resource Description Framework which extends XML. RDF was designed for describing information that can be readable and understandable by computer applications [1]. Some of the usages of RDF can be sampled as follows;

Describing,

- content for search engines,
- properties for shopping items, such as price and availability,
- information about web pages, such as content, author, created and modified date,
- content and rating for web pictures,

RDF includes explanation of Resource, Property, and Property value:

- A Resource could be anything that can have a URI, for example "http://www.abc.com/XYZ"
- A Property is a Resource that has a name, such as "cd" or "homepage"
- A Property value is the value of a Property, such as "Pop Music" or "http://www.abc.com" (here we can see that a property value can be another resource)

The following RDF document describes the resource “http://www.abc.com/XYZ":

```
<?xml version="1.0"?>
<RDF>
  <Description about="http://www.abc.com/XYZ">
    <cd>Pop Music</cd>
    <homepage>http://www.abc.com</homepage>
  </Description>
</RDF>
```

RDF provides a composable and extensible standard when building data models. For this purpose, RDF needs a layer named RDF Schema that allows a designer to define and publish the vocabulary used in their RDF data model also based on XML [1] [2] [34] [39]. To represent machine-interpretable knowledge, some Semantic Web applications use RDF.

3.3.5 OWL

OWL refers to Web Ontology Language which is written in XML. OWL is designed for only applications that need to process information in a document. Ontology includes representation of terms and their interrelationships, and OWL is more expressive when compared with XML, RDF, and RDF-S in terms of representing the meaning of terms in vocabularies and the relationships between those terms [1] [2] [19].

OWL has three sublanguages designed for use by specific communities of developers and users:

- OWL Lite
- OWL DL (includes OWL Lite)
- OWL Full (includes OWL DL)

"OWL is a component of the Semantic Web activity", and OWL became a W3C (World Wide Web Consortium) Recommendation in February 2004 [1] [19].

3.4 Previously Developed Semantic Web Applications

OntoSem: stands for Ontological Semantics which is a language-independent ontology [35].

Hakia: is a general purpose Semantic Search engine via Natural Language Interface for English and Turkish [12].

Foaf: is a Semantic Web application to share and use information among people where the data is stored in RDF [36].

Simile: stands for Semantic Interoperability of Metadata and Information in unlike Environments. SIMILE project is directed by MIT libraries and focuses on improving interoperability among ontologies, vocabularies, digital entities and meta data [37].

Sioc: Semantically-Interlinked Online Communities provides the main concepts and properties required to describe information from online communities such as weblogs, mailing lists and image galleries [38].

PANTO: stands for Portable nAtural laNguage inTerface to Ontologies, which provides users about acquiring the needed information from formally defined ontologies [13].

4. DESIGN OF THE SYSTEM

4.1 Domain

The aim of this thesis is to create a Turkish natural language interface for an academic Semantic Web. The domain of the system contains information about Turkish universities such as location, staff, contingencies, entrance exam scores, etc. which will help a student in making his/her preferences for a major after taking the university exam.

The system is a prototype and developed especially for Turkish high school students and university candidates who are novice users with little knowledge about computer technologies.

Currently, the domain of the system covers two departments of two universities; namely Atılım and Pamukkale.

4.2 General Structure of the System

The components of the system are shown in Figure 4.1.

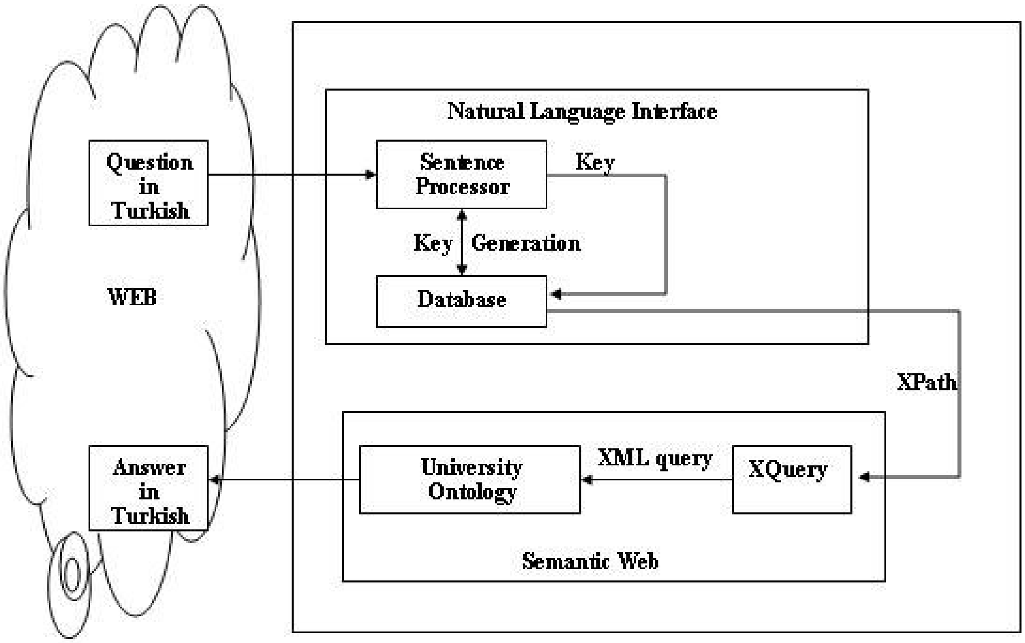


Figure 4.1 Components of the System

As shown in Figure 4.1, the question written in Turkish is inputted to the system using a Natural Language Interface. The inputted question is sent to the Sentence Processor which analyzes the words morphologically, and splits the sentence into words. Then, the Sentence Processor starts the key-generation by retrieving word-specific information from the Database. After syntactic analysis, the keys are combined to form the sentential key. Afterwards, the sentential key is translated into the XPath information and is sent to the Semantic Web component of the system. The XQuery service receives the XPath to form an XML query which retrieves the answer of original question, which is then outputted.

The modules of the system are described as follows;

Sentence Processor module makes morphological analysis of words using the word-stemming approach and generates the keys with the interaction of database. Firstly, Sentence Processor splits the question sentence into words, then tries to find a match with the roots of the words (i.e. word stemming) included in the lexicon of the system which is stored in the database component. For each word match, a specific number is added to the sentential key during the Key Generation process.

In Figures 4.2 and 4.3, the process of Key Generation is shown on two examples using Mealy machines;

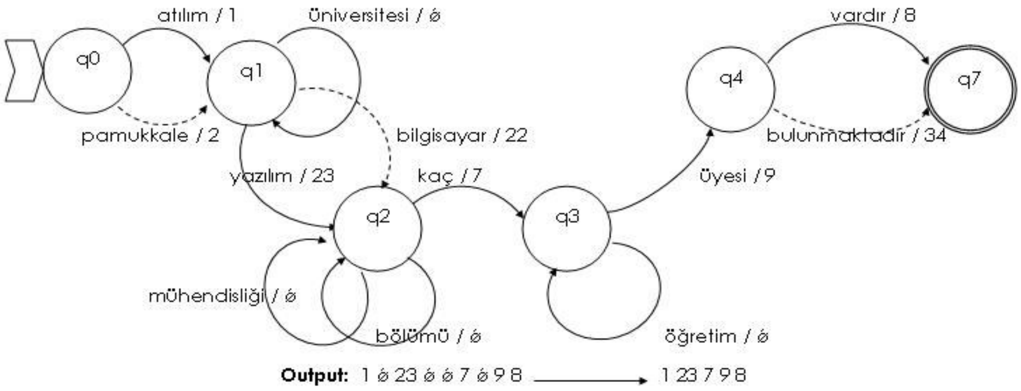


Figure 4.2 Mealy machine for the question "Atılım üniversitesi yazılım mühendisliği bölümünde kaç öğretim üyesi vardır?"

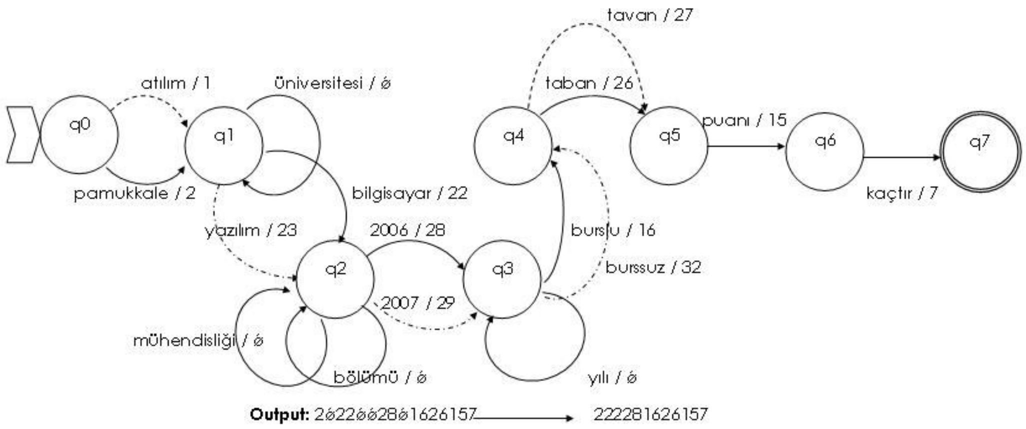


Figure 4.3 Mealy machine for the question "Pamukkale üniversitesi bilgisayar mühendisliği bölümünün 2006 yılı burslu taban puanı kaçtır?"

The Sentence Processor also handles the free-word order property of Turkish Language. Consequently, even if the user changes the word order of a specific question, the sentence processor is able to generate the same key.

The Figures 4.4 and 4.5 show examples of Mealy machines for questions with free-word-order variations;

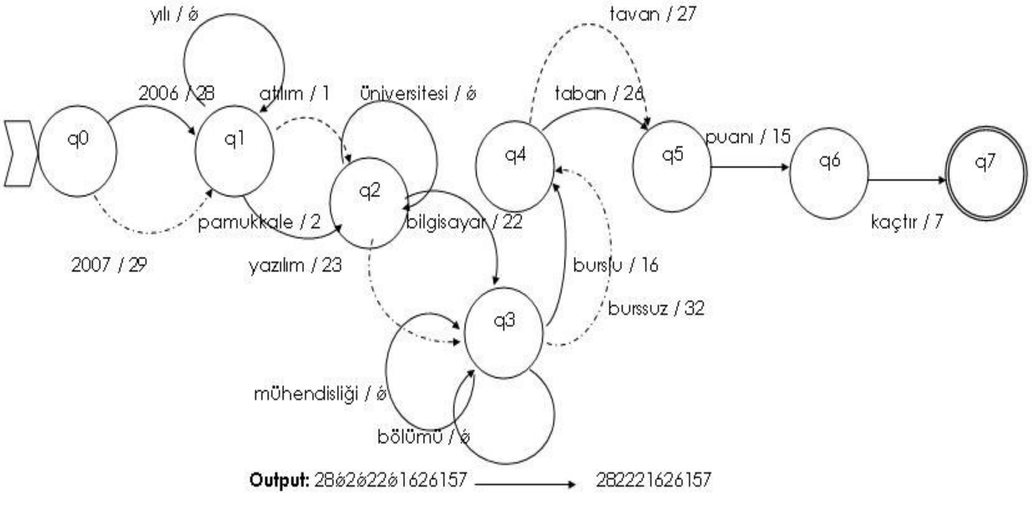


Figure 4.4 Mealy machine for the question "2006 yılında pamukkale üniversitesi bilgisayar mühendisliği bölümü burslu taban puanı kaçtır?"

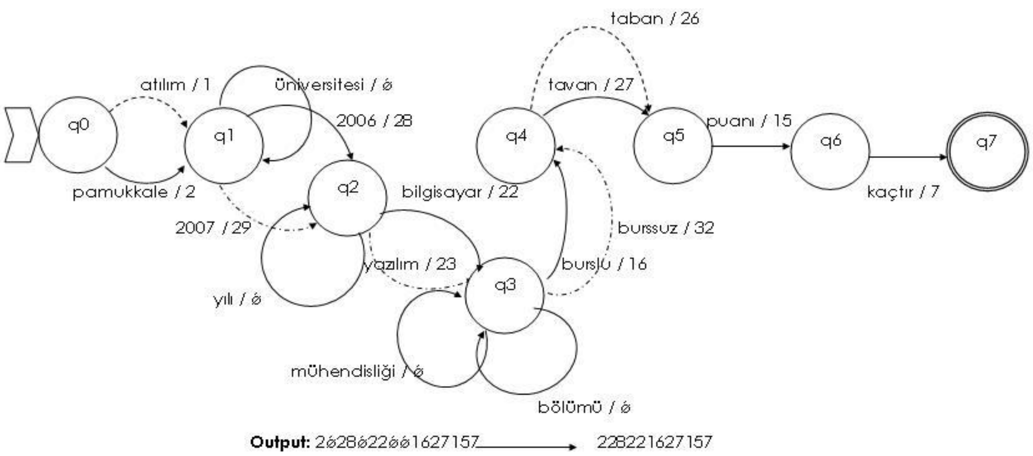


Figure 4.5 Mealy machine for the question "pamukkale üniversitesi 2006 yılındaki bilgisayar mühendisliği burslu tavan puan kaçtır?"

As a last action for the Sentence Processor, the Sentential Key is translated into the corresponding XPath with the help of the Database which is then sent to XQuery module.

The Database module of the system is a Relational Database including two tables for key generation.

The first table is used in the morphological analysis as the lexicon in the word-stemming process by the Sentence Processor in Key Generation. The second table is used in the sentential key to XPath translation. The corresponding XPath is generated by matching the id field of this table with the previously generated key.

The XQuery module inputs the XPath and generates the XML query to retrieve the answer from the University Ontology

Finally, *The University Ontology* module stores the yearly information about some of the Universities in Turkey dealing with the staff, contingencies, scores, publications and location. As the last step in the system, based on the XML query, the University Ontology module generates the final answer.

5. IMPLEMENTATION

5.1 Technical specifications

The natural language interface developed for this thesis is a cross-platform and platform-independent application. The technical specifications of the system are described in detail in this section. Firstly, in the Web interface of the system, HTML is used to retrieve the inputted question via a text box which is then sent to the Sentence Processor module of the system.

In the development of the Sentence Processor, Java technology is used for syntactic and morphological analysis of a given question. Java is also used to handle the connection to the Database module. The reasons for choosing Java technology are that Java technology is a programming environment which is object-oriented, platform-independent with multithreading capabilities. Java provides the basis for the Web and networked services, applications and also it supports cross-platform applications through other technologies used in computer applications [14].

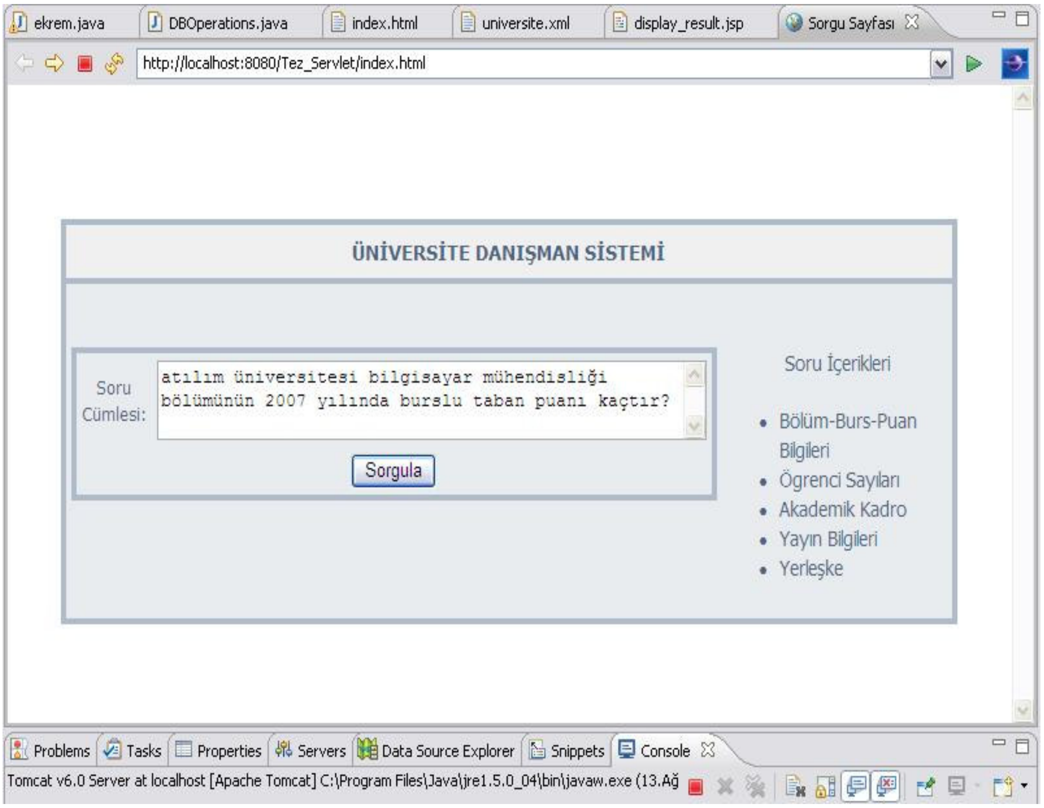
In the Database module, the relational database model is used. In the development of the database, Phpmyadmin tool is utilized, since it provides flexible usage in the database technologies. Phpmyadmin is a tool written in PHP language developed to handle the administration of MySQL over the Web which can also execute any SQL statements, manage keys on fields, manage privileges, and export data into various formats. Phpmyadmin tool supports 55 languages including Turkish [15].

In the development of the Xquery module, JavaServer Pages (JSP) technology is used since it provides simplified and quick way to create dynamic web content. JSP is included in the Java technology family, providing rapid development of Web-based applications which are platform independent.

For the University Ontology module, XML technology is used in the creation of the ontology. The ontology is created manually including information about selected Turkish Universities such as quota information for scholarship and non-scholarship students, number of enrolled students, location of the universities, number of staff and number of publications on a yearly basis. Finally in the Answer page, JSP technology is used for answering the question by displaying the extracted answer from the University Ontology Module where the content of this page is dynamically written by JSP in the format of HTML (see Appendix *B*).

5.2 Execution of the System

The user inputs the question in Turkish into the Web Page as shown in Figure 5.2.1;



Soru Cümlesi: atılım üniversitesi bilgisayar mühendisliği bölümünün 2007 yılında burslu taban puanı kaçtır?

```
Tomcat v6.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.5.0_04\bin\javaw.exe (13.Ağ
```

Figure 5.2.1 Input Web Page

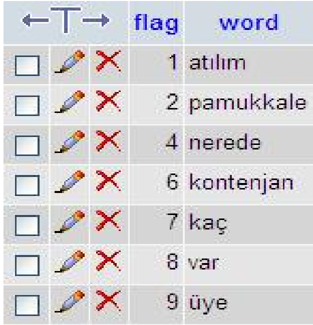
The inputted question is transferred into "display_result.jsp" file as a string argument when "Sorgula" button is clicked (See Appendix *A*). Following this action, Natural Language Interface module begins its process. In the JSP file, an object is created from a Java class, and then the result is retrieved through the methods defined in these class objects. In the Java-progress of the system, two java classes named "ekrem" and "DBOperations" have been utilized.

With the inputted string, the method named "Sorgula" is called via object named "e" which is created from class "ekrem.java" (See Appendix *C*). The method "Sorgula" splits the string with the predefined method in Java named StringTokenizer, into words.

During the tokenization process, some delimiters such as the blank character and other punctuation symbols (", - ?", etc.) are removed.After the tokenization process, each of the words in the question are inserted dynamically into an object named “v” which is created from class “Vector” in Java class library.

At the same time, database connection is established via an object named wordMap which is created from the class “DBOperations.java” (See Appendix *D*). The class DBOperations establishes the database connection and has the function named getWordsInDB which receives an SQL statement and returns resulting database entry as in the object of the class Map. The SQL query is sent to the database which is used as the lexicon and the content of table named “kelime” is retrieved into an object namely “wordMap” which is created from class “Map”.

Some of the entries of table named “kelime” in the database are shown in Figure 5.2.2.

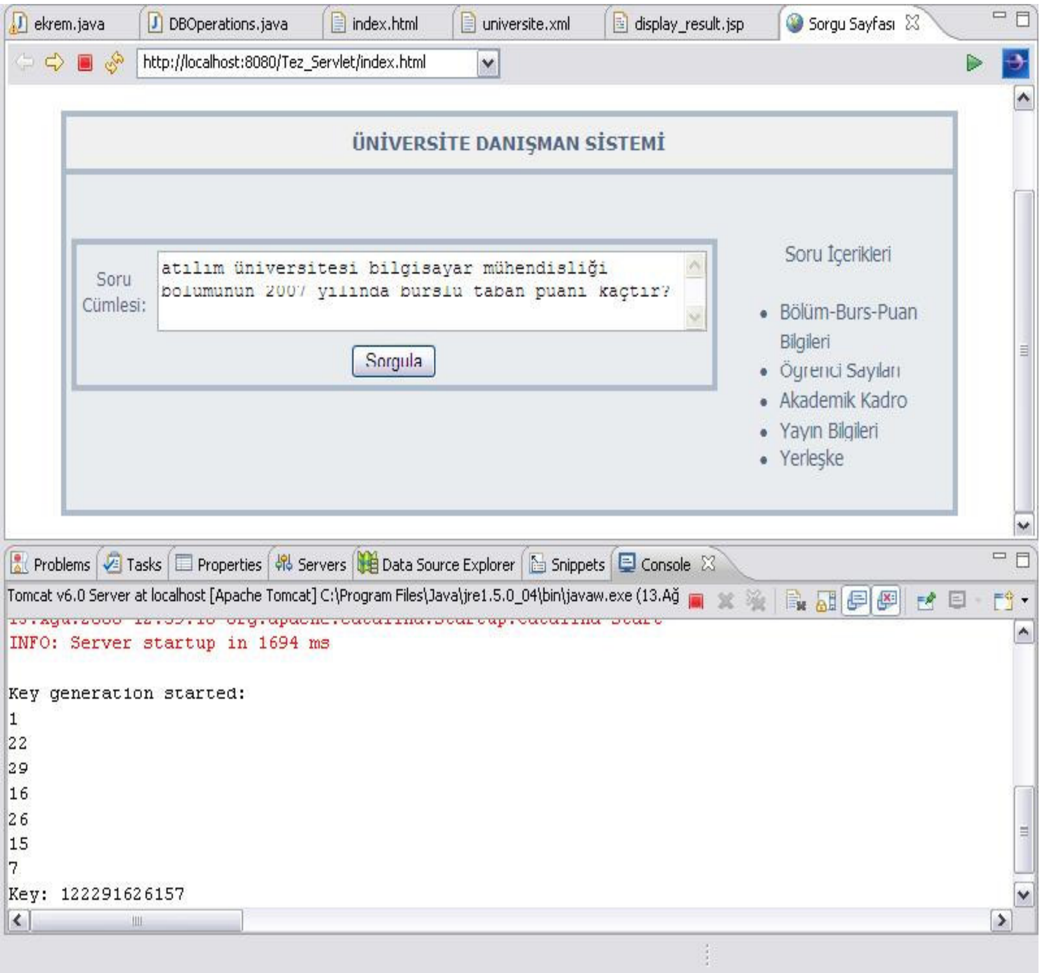


←T→			flag	word
□	✎	✕	1	atılım
□	✎	✕	2	pamukkale
□	✎	✕	4	nerede
□	✎	✕	6	kontenjan
□	✎	✕	7	kaç
□	✎	✕	8	var
□	✎	✕	9	üye

Figure 5.2.2 Database table, “kelime”

After these steps, The Key Generation process starts. For each word in the question that has been inserted into Vector object “v” in the Tokenization process, The Key Generation Process attempts to find a match. While finding a match, each word is compared with the object “wordMap’s” entries in column “word” retrieved by the Database module, in column “word” of table “kelime” which includes the root of the words that can possibly be used in the questions.

When a match is caught, a sentential key from column “flag” is added to the resulting key variable “flagCount” as shown in Figure 5.2.3.



Soru Cümlesi: atılım üniversitesi bilgisayar mühendisliği bolumunun 2007 yılında burslu taban puanı kaçtır?

- Bölüm-Burs-Puan Bilgileri
- Öğrenci Sayıları
- Akademik Kadro
- Yayın Bilgileri
- Yerleşke

Tomcat v6.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.5.0_04\bin\javaw.exe (13.Ağ

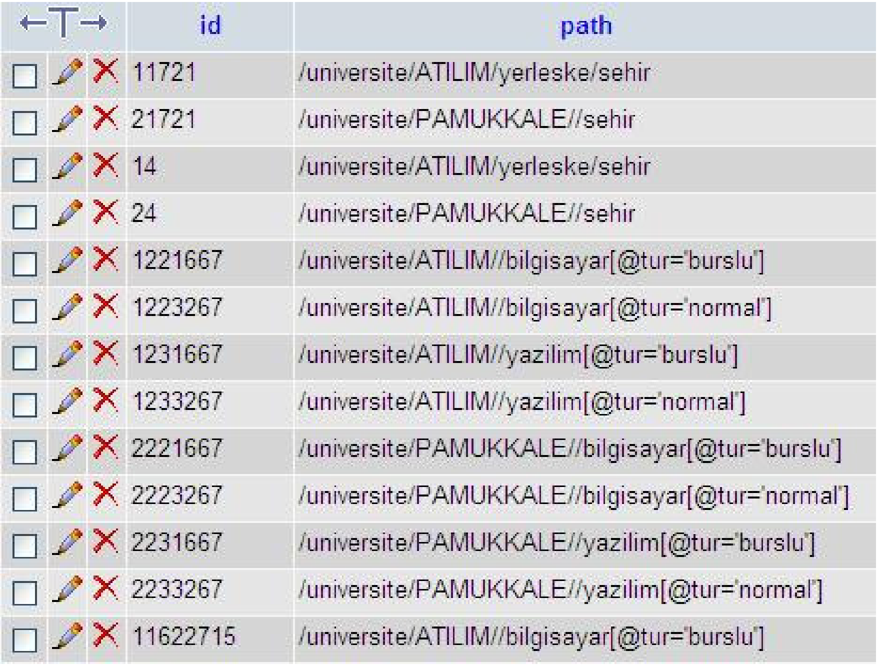
```
INFO: Server startup in 1694 ms

Key generation started:
1
22
29
16
26
15
7
Key: 122291626157
```

Figure 5.2.3 Key Generation Process

After The Key Generation is completed, the resulting Key is created according to words and their order in the question. Since the Turkish language has free sentence order, the system gives answer to seven question templates with 340 variations of different word orders resulting in the same Xpath value.

After the Key is evaluated, function getWordsInDB is called again with a new SQL statement resulting in the whole table namely "path" in the Database module which is stored into a new object named "worMap2" created from class Map.

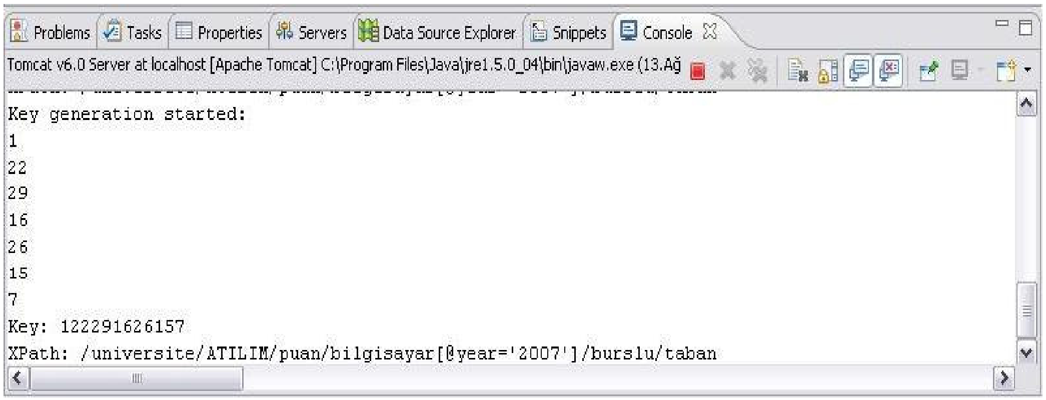
Some of the entries of table named "path" in the database are shown in Figure 5.2.4.



←T→	id	path
□ ✎ ✗	11721	/universite/ATILIM/yerleske/sehir
□ ✎ ✗	21721	/universite/PAMUKKALE//sehir
□ ✎ ✗	14	/universite/ATILIM/yerleske/sehir
□ ✎ ✗	24	/universite/PAMUKKALE//sehir
□ ✎ ✗	1221667	/universite/ATILIM//bilgisayar[@tur='burslu']
□ ✎ ✗	1223267	/universite/ATILIM//bilgisayar[@tur='normal']
□ ✎ ✗	1231667	/universite/ATILIM//yazilim[@tur='burslu']
□ ✎ ✗	1233267	/universite/ATILIM//yazilim[@tur='normal']
□ ✎ ✗	2221667	/universite/PAMUKKALE//bilgisayar[@tur='burslu']
□ ✎ ✗	2223267	/universite/PAMUKKALE//bilgisayar[@tur='normal']
□ ✎ ✗	2231667	/universite/PAMUKKALE//yazilim[@tur='burslu']
□ ✎ ✗	2233267	/universite/PAMUKKALE//yazilim[@tur='normal']
□ ✎ ✗	11622715	/universite/ATILIM//bilgisayar[@tur='burslu']

Figure 5.2.4 Database table, "path"

At this point, the system starts to find a match for the Key by traversing the entries in the object "wordMap" in column "id " to reach the resulting Xpath value from column "path". Figure 5.2.5 shows the situation of finding a match.



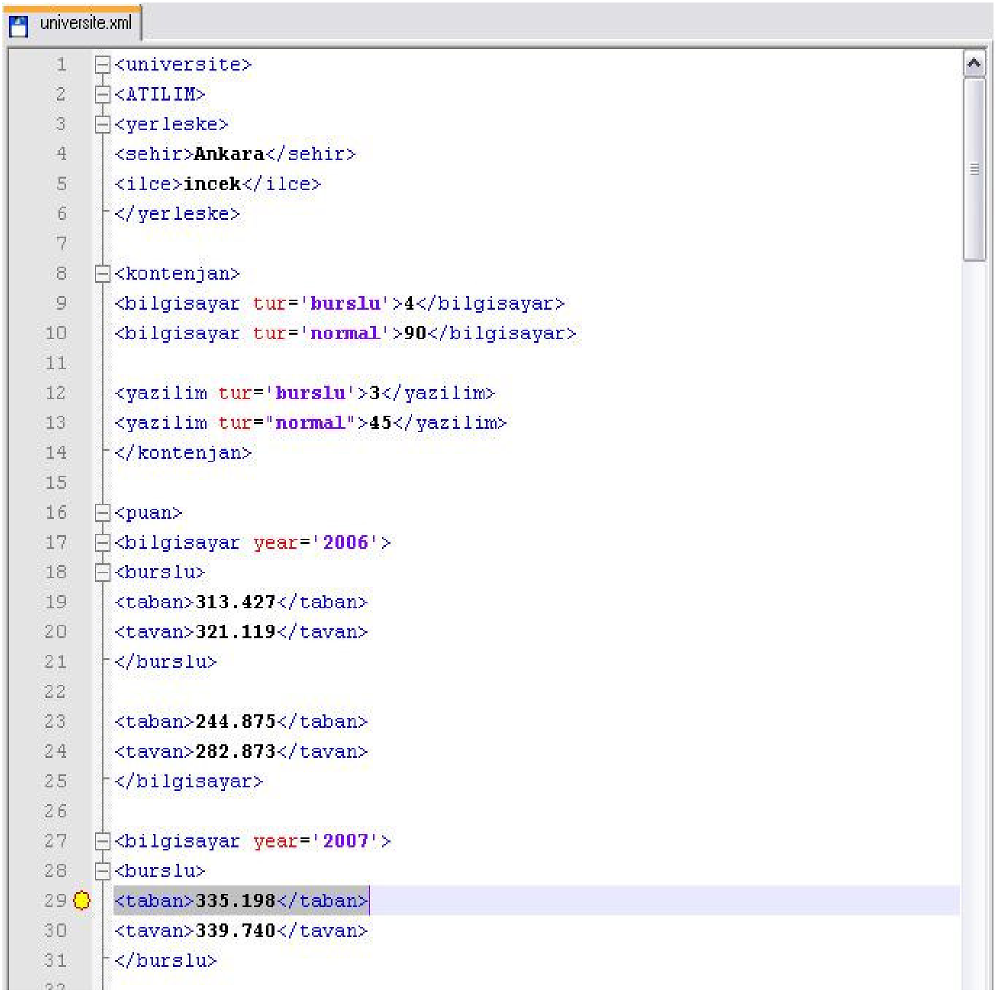
Tomcat v6.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.5.0_04\bin\javaw.exe (13.Ağ

```
Key generation started:
1
22
29
16
26
15
7
Key: 122291626157
XPath: /universite/ATILIM/puan/bilgisayar[@year='2007']/burslu/taban
```

Figure 5.2.5 A Sample of the Key match

The resulting Xpath value is returned to variable named “cevap” located in file “display_result.jsp”.

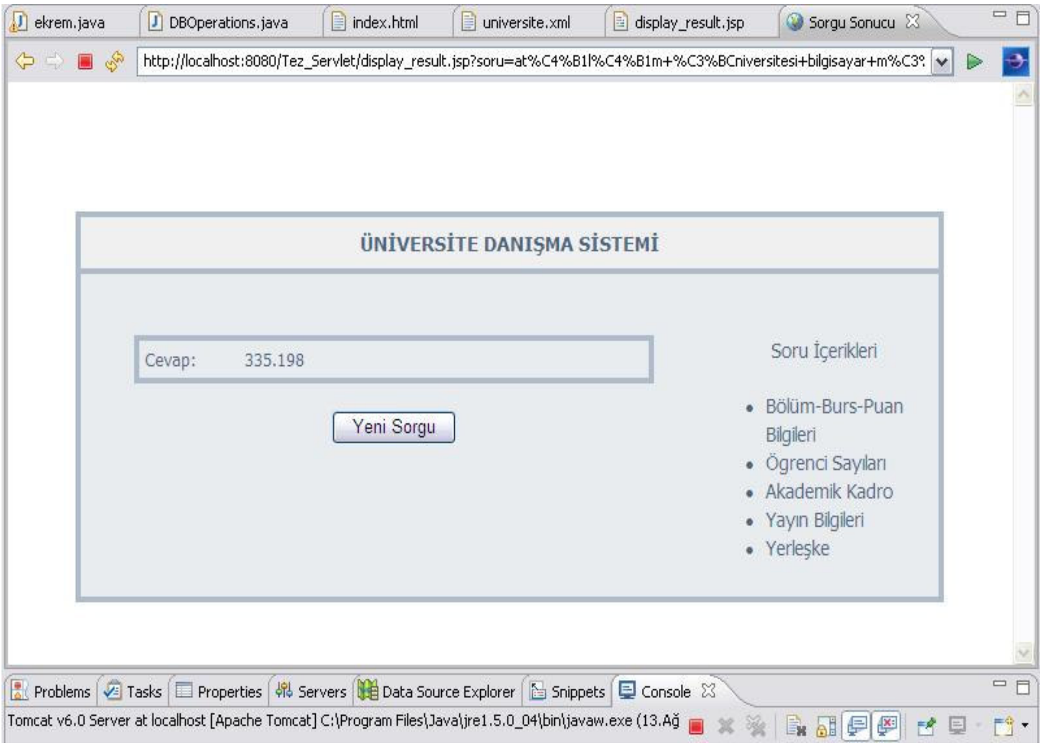
After resulting Xpath value is included in “cevap”, the XML file named “universite.xml” is loaded into the JSP file with function “loadXMLdoc()”. And the XQuery part of the system reaches the resulting node in the XML file with the resulting answer as shown in Figure 5.2.6.



```
<universite>
<ATILIM>
<yerleske>
<sehir>Ankara</sehir>
<ilce>incek</ilce>
</yerleske>

<kontenjan>
<bilgisayar tur='burslu'>4</bilgisayar>
<bilgisayar tur='normal'>90</bilgisayar>

<yazilim tur='burslu'>3</yazilim>
<yazilim tur="normal">45</yazilim>
</kontenjan>

<puan>
<bilgisayar year='2006'>
<burslu>
<taban>313.427</taban>
<tavan>321.119</tavan>
</burslu>

<taban>244.875</taban>
<tavan>282.873</tavan>
</bilgisayar>

<bilgisayar year='2007'>
<burslu>
<taban>335.198</taban>
<tavan>339.740</tavan>
</burslu>
```

Figure 5.2.6 Content of the ontology file “universite.xml”

The answer of the system is shown in Figure 5.2.7.



http://localhost:8080/Tez_Servlet/display_result.jsp?soru=at%C4%B1l%C4%B1m+%C3%BCniversitesi+bilgisayar+m%C3%

Cevap: 335.198

```
Tomcat v6.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.5.0_04\bin\javaw.exe (13.Ağ
```

Figure 5.2.7 Answer Page

The same question with different word orders resulting in the same XPath value is sampled as follows;

1. “2007 yılında atılım üniversitesi bilgisayar mühendisliği bölümü burslu taban puanı kaçtır?”
2. “2007 yılında atılım üniversitesi bilgisayar mühendisliği bölümü burslu puanı kaçtır?”
3. “atılım üniversitesi 2007 yılındaki burslu bilgisayar mühendisliği bölümü puanı kaçtır?”
4. “atılım üniversitesi 2007 yılındaki bilgisayar mühendisliği bölümü burslu puanı kaçtır?”

5.3 Evaluation of the system

For evaluation purposes, the system stores the information for two Turkish universities namely Atılım University and Pamukkale University. The departments covered in the ontology are Department of Computer Engineering and Department of Software Engineering. The topics that can be questioned by the system can be listed under the following headings;

- Location of the University,
- Scholarship – Score information,
- Floor – Ceiling Scores,
- Number of Students in the Departments,
- Academic personnel,
- Publications

The system includes seven basic templates for Turkish questions. The templates include 340 variations of the questions to handle the free-word-order property of Turkish language. In the lexicon located in the Database module of the system, there are 30 words which are the roots of the words used in the Key generation module of the system.

The system was evaluated by five people for fifty sentences. The results are given in the Table 1;

Table 1: Evaluation results

Test number	Number of Correct Answers	Number of Unanswered Questions	Total
1	7	3	10
2	8	2	10
3	6	4	10
4	5	5	10
5	7	3	10
Total	**33**	**17**	**50**

As seen in the Table 1, the system is working with 66% success rate according to the test results. Some of the questions in testing phase are shown in Appendix *E*.

To increase the success ratio of the system,

- The university coverage,
- The number of departments covered,
- The number of question templates,
- The number of words in the lexicon,
- Variations for the question templates for the free-word-order property of the Turkish language,

could be increased.

6. CONCLUSION

The target of this thesis was to develop a Turkish Natural Language Processing Interface for an academic Semantic Web. The main purpose underlying the Semantic Web technology is to keep the required information in a computer-understandable format and for the Natural Language Processing Interface is to increase the communication power in terms of human-computer interaction.

The system is developed especially for Turkish high school students and university candidates who are novice users with little knowledge about computer technologies. For this reason, domain of the Ontology includes information about selected Turkish Universities whereas the lexicon includes keywords related to the possible questions that can be asked by the users of the system. In addition, the database used by the system stores the information about possible question templates. For the questions, some restrictions are made in terms of the number of question templates and the variations of questions in a template due to free-word-order in Turkish language.

Since the system is a prototype, the following improvements could be made as future work to increase the success rate of the system.

The Web interface of the system could be improved in terms of the usability issues of a Web page i.e. readability, accessibility and internationalization.

The Natural Language Interface module could be improved by expanding the database to find matches to different questions parsed by the Parser module and additional keywords could be inserted into the lexicon again to cover new question topics about Turkish universities. The ontology can also be expanded to cover more departments and universities in Turkey.

For this thesis, the Ontology module is created manually and as a future work, an automatic or semi-automatic knowledge acquisition system for the Semantic Web module could be added to access current information about the Turkish Universities and update the present information in the Ontology.

During the implementation phase of the system, the main problem was deciding the possible set of unordered question variations because the possible user profile of the system is novice users with little knowledge about computers. To meet the requirement aspects of the system, the important subjects were chosen about the Turkish universities to make it easy for the users to reach the relevant information. The reasons for developing a Turkish language interface for the system is that a Turkish university candidate who is a novice computer user is able to express himself/herself better in his/her native language.

The developed system can be used for other domains with the flexibility of changing the lexicon related to the new domain, and also adding new keys that are possible to extract from the questions related to the new domain of the knowledge.

Introduced as the future of the Internet, the Semantic Web keeps information and the knowledge in a machine-understandable form and provides a more declarative, flexible and powerful information retrieval mechanism to be used for different knowledge domains.

Nowadays, the current trends in computer technologies and the improvements of the Semantic Web technologies will cause an increase in the usage of Natural Language Interfaces to meet the user requirements in computer technologies to enhance effectiveness of the new systems.

7. REFERENCES

[1] *W3Schools Online Web Tutorials*. [Accessed June 2008]. Available from World Wide Web :< http://www.w3schools.com>

[2] Baader, F, Horrocks, I & Sattler, U. 2005. Description Logics as Ontology Languages for the Semantic Web. *Lecture Notes in Computer Science.* pp. 228 – 248.

[3] Oflazer, K, Bozşahin , C, 2006. Natural Language Processing in Turkish. Available from World Wide Web: <http://turkoloji.cu.edu.tr/DILBILIM/turkce_dogal_dil_isleme.pdf>

[4] Bose, R, 2004. Natural Language Processing: Current state and future directions. *Proceedings of the International Journal of the Computer, the Internet and Management (IJCIM), Vol.12#1(January – April, 2004).* pp. 1 – 11.

[5] Özcan, E, Şeker, Ş E, Karadeniz, Z İ, 2004. Generating Java Class Skeleton Using A Natural Language Interface, NLUCS - 2004, *In conjunction with International Conference on Enterprise Information Systems (ICEIS) 2004,* pp. 126-134*, Portugal.*

[6] Say, A C C., Çetinoğlu, Ö, Demir, Ş, Öğün, F. 2004. A Natural Language Processing Infrastructure for Turkish. *In Proceedings of COLING 2004, August 2004, Geneva, Switzerland.*

[7] Erguvanlı, E, E 1984. The Function of Word Order in Turkish Grammar. *Proceedings of the University of California Publications in Linguistics Berkeley, Cal. 1984, vol. 106*. pp. 1-179 (4 p.)

[8] *SHRDLU* Project. [Accessed July 2008]. Available from World Wide Web : < http://hci.stanford.edu/~winograd/shrdlu/>

[9] Çetinoğlu, Ö, 2001. *A Prolog Based Natural Language Processing Infrastructure for Turkish*, M.S. Theses, Department of Computer Engineering, Boğaziçi University.

[10] Şeker, Ş. E., 2003. *Design and Implementation of a Personal Calender with a Natural Language Interface in Turkish.* M. S. Thesis. Department of Computer Engineering, Yeditepe University.

[11] Maden, İ, Demir, Ş, Özcan, E, 2003. *Turkish Natural Language Interface for Generating SQL Queries.* In “TBD Ulusal Bilişim Kurultayı”.

[12] *Hakia* project [Accessed July 2008]. Available from World Wide Web: < http://www.hakia.com.>

[13] Wang, C, Xiong, M, Zhou, Q, Yong, Y, 2007. A Portable Natural Language Interface to Ontologies. *Proceedings of The 4th European Semantic Web Conference.* [Accessed July 2008]. Available from World Wide Web: <http:// apex.sjtu.edu.cn>

[14] *Sun Microsystems*. [Accessed July 2008]. Available from World Wide Web :<http://www.sun.com/java/>

[15] *Php My Admin Home Page*. [Accessed July 2008]. Available from World Wide Web: <http://www.phpmyadmin.net/home_page/index.php>

[16] Davies, J, Studer, R, Warren, P. 2006. *Semantic Web Technologies - trends and research in ontology based systems*. England: John Wiley & Sons Ltd.

[17] Lyman P, Hal R. Varian. 2003. *How much information?.* Technical Report, UC Berkeley. Available from World Wide Web : <http://www.sims.berkeley.edu/research/projects/how-much-info-2003/>

[18] Ding, Y, Rijsbergen, C J V, Ounis, I, and Jose, J M. 2003. Report on ACM SIGIR Workshop on 'Semantic Web' SWIR 2003. *In Proceedings of SIGIR Forum 2003*. pp 45-49.

[19] *World Wide Web Consortium.* [Accessed June 2008]. Available from World Wide Web :< http://www.w3.org>

[20] Ding Y, Fensel, D. Klein, M & Omelyenko, B. 2002. 'The Semantic Web: Yet Another Hip?'. *Data and Knowledge Engineering.* pp.205 – 227.

[21] Berners-Lee, T, Hendler, J & Lassila, O. 2001. The Semantic Web. *Scientific American*. Pp. 35 – 43.

[22] Gruber T. 1993. *A Translation Approach to Portable Ontologies*. Knowledge Acquisition 5(2):199–220. Available from World Wide Web : <http://ksl-eb.stanford.edu/KSL_Abstracts/KSL-92-71.html>

[23] Zhang, J. 2007. Ontology and The Semantic Web. *Proceedings of the North American Symposium on Knowledge Organization.* pp. 9 – 20.

[24] Silva, N & Rocha, J. 2003. Semantic Web Complex Ontology Mapping. *IEEE/WIC International Conference on Web Intelligence.* pp. 82 – 88.

[25] Man, L, Xiao-Yong, D & Shan, W. 2005 Learning Ontology from Relational Database. *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics.* 6. pp. 3410 – 3415.

[26] Asuncion, G P. Corcho, O. 2002. Ontology Languages for the Semantic Web. *IEEE Intelligent Systems.* 17 (1). pp . 54 – 60.

[27] Jacob, E K. 2003. Ontologies and the Semantic Web. *Bulletin of the American Society for Information Science and Technology.* 29 (4). pp. 19 – 22.

[28] Elmasri, R & Navathe, S B. 2003. *Fundamentals of Database Systems.* California: Pearson/Addison Wesley.

[29] Gray, P M D, Kulkarni, K G & Paton, N W. 1992. *Object Oriented Databases: A Semantic Data Model Approach.* New Jersey: Prentice Hall.

[30] Asunción Gómez-Pérez, Mariano Fernández-López and Oscar Corcho, 2004. *Ontological Engineering - with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*. London: Springer: ISBN 1-85233-551-3

[31] Staab S, Studer R (Eds). 2004. *Handbook on Ontologies*. International Handbooks on Information Systems. London: Springer: ISBN 3-540-40834-7.

[32] Ray, E T. 2003. *Learning XML.* California: O'Reilly.

[33] Heflin, J & Hendler, J. 2000. Dynamic Ontologies on the Web. *Proceedings of the Seventeenth National Conference on Artificial Intelligence.* pp. 443 – 449.

[34] Hjlem, j. 2001. Creating the Semantic Web with RDF. New York: Wiley.

[35] Nirenburg, S., and V. Raskin. (2001). Ontological Semantics, Formal Ontology and Ambiguity. *Proceedings of the Second International Conference on Formal Ontology in Information Systems.* Ogunquit, ME, October, pp. 151-161.

[36] *Foaf* Project. [Accessed July 2008]. Available from World Wide Web :<http://www.foaf-project.org>

[37] *Simile* Project. [Accessed July 2008]. Available from World Wide Web :<http://simile.mit.edu>

[38] *sioc-project.org* | S*emantically Interlinked Online communities.* [Accessed July 2008]. Available from World Wide Web :< http://sioc-project.org>

[39] HP Laboratories. [Accessed July 2008]. Available from World Wide Web :<http://www.hpl.hp.com/semweb/sw-technology2.htm>

8. APPENDIX

Appendix A

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Sorgu Sayfası</title>
<link rel="stylesheet" type="text/css"
href="smoothtaste.css" />
</head>
<body>
  <p>&nbsp;</p>
  <p>&nbsp;</p>
<form action="display_result.jsp" method="get">
  <table style= width="600" border="0" align="center">
  <tr>
    <th colspan="2" scope="col">&Uuml;N&#304;VERS&#304;TE DANI&#350;MAN
S&#304;STEM&#304;</th>
  </tr>
  <tr>
    <td width="475" height="219" align="center" valign="top">
              <p>&nbsp;</p>
              <table align="center">
            <tr>
                        <td>Soru Cümlesi:</td>
                        <td><textarea name="soru" cols="50" rows="3"></textarea></td>
            </tr>
            <tr>
                        <td colspan="2" align="center"><input name="submit" type=submit
value="Sorgula"></td>
            </tr>
      </table>
             </td>
    <td width="169" valign="top"><p align="center">&nbsp;</p>
      <p align="center">Soru İçerikleri </p>
      <ul>
        <li>Bölüm-Burs-Puan Bilgileri </li>
        <li>Ögrenci Sayıları </li>
        <li>Akademik Kadro</li>
        <li>Yayın Bilgileri</li>
        <li>Yerleşke</li>
      </ul>      </td>
  </tr>
</table>
```

</form></body></html>

Appendix B

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>

<%@page import="Ekrem.ekrem" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Sorgu Sonucu</title>
<link rel="stylesheet" type="text/css"
href="smoothtaste.css" />
</head>
<body>
<%!
        Ekrem.ekrem e = new Ekrem.ekrem();
        String soru;
        String cevap;
%>
<%
        soru = request.getParameter("soru");
        cevap = e.Sorgula(soru);
        System.out.print("XPath: "+cevap);
%>
  <p>&nbsp;</p>
  <p>&nbsp;</p>
  <table style= width="600" border="0" align="center">
    <tr>
      <th colspan="2" scope="col">&Uuml;N&#304;VERS&#304;TE DANI&#350;MAN
S&#304;STEM&#304; </th>
    </tr>
    <tr>
      <td width="475" height="219" align="center" valign="top"><p>&nbsp;</p>
          <table width="400" align="center">
            <tr align="left">
              <td width="70">Cevap:</td>
              <td width="312"><script type="text/javascript">
function loadXMLDoc(fname)
{
var xmlDoc;
// code for IE
if (window.ActiveXObject)
  {
  xmlDoc=new ActiveXObject("Microsoft.XMLDOM");
  }
// code for Mozilla, Firefox, Opera, etc.
else if (document.implementation && document.implementation.createDocument)
  {
  xmlDoc=document.implementation.createDocument("","",null);
  }
else
  {
  alert('Your browser cannot handle this script');
  }
xmlDoc.async=false;
xmlDoc.load(fname);
return(xmlDoc);
}

        xml=loadXMLDoc("universite.xml");
        path="<%=cevap%>";
        //document.write(path);
        // code for IE
        if (window.ActiveXObject)
        {
        var nodes=xml.selectNodes(path);

        for (i=0;i<nodes.length;i++)
          {
//        document.write("Cevap: ");
          document.write(nodes[i].childNodes[0].nodeValue);
          document.write("<br />");
          }
        }
        // code for Mozilla, Firefox, Opera, etc.
        else if (document.implementation && document.implementation.createDocument)
        {
        var nodes=document.evaluate(path, xml, null, XPathResult.ANY_TYPE, null);
        var result=nodes.iterateNext();

        while (result)
          {
          document.write(result.childNodes[0].nodeValue);
          document.write("<br />");
          result=nodes.iterateNext();
          }
        }

function yonlendir()
{
window.location="index.html";
}
   </script>              </td>
        </tr>
       </table>
```

```
    <p>
      <input name="button" type="button" onClick="yonlendir();" value="Yeni Sorgu">
    </p></td>
   <td width="169" valign="top"><p>&nbsp;</p>
     <p align="center">Soru &#304;&ccedil;erikleri </p>
     <ul>
      <li>B&ouml;l&uuml;m-Burs-Puan Bilgileri </li>
      <li>&Ouml;grenci Say&#305;lar&#305; </li>
      <li>Akademik Kadro</li>
      <li>Yay&#305;n Bilgileri</li>
      <li>Yerle&#351;ke</li>
     </ul></td>
  </tr>
 </table>
</body>
</html>
```

Appendix C

```
package Ekrem;


import java.util.*;
public class ekrem {

    public String Sorgula(String soru)
    {
        String result = "";
        String delimiters = " -?";
        StringTokenizer stoken;
        Vector<String> v = new Vector<String>();

        int i = 0;
        String tmp = soru;

        stoken = new StringTokenizer(tmp, delimiters);
        DBOperations dbOp = new DBOperations();

        String sql = "select * from kelime";
        Map wordMap = dbOp.getWordsInDB(sql);

        while (stoken.hasMoreTokens()) {
            v.addElement(stoken.nextToken().toString());
        }

        String flagCount = "";
        System.out.println("\nKey generation started:");
        for (i = 0; i < v.size(); i++) {

            for (Iterator iterator = wordMap.keySet().iterator(); iterator.hasNext();)
{
                String word = (String) iterator.next();
                if (v.elementAt(i).startsWith(word)) {
                    Integer flagInt = (Integer) wordMap.get(word);
                    System.out.println(flagInt);
                    flagCount = flagCount + "" + flagInt;

                }
            }

        }

        System.out.println("Key: "+flagCount);
        sql = "select * from abc";
        Map wordMap2 = dbOp.getWordsInDB2(sql);
        for (Iterator iterator2 = wordMap2.keySet().iterator(); iterator2
                .hasNext();) {

            String word = (String)iterator2.next();
            if (flagCount.trim().equals(word)) {
                result = (String) wordMap2.get(word);
                return result;
            }
        }
        System.out.print("XPath: "+result);
        return "sonuc bulunamadi";
    }
}
```

Appendix D

```
package Ekrem;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.HashMap;
import java.util.Map;
import com.mysql.*;
public class DBOperations {

    public DBOperations() {

    }

    Map getWordsInDB(String sql){
        Map wordMap = new HashMap();

        //Loar driver
        String driverName = "com.mysql.jdbc.Driver";
        try {
            Class.forName(driverName);

            //URL
            String URL = "jdbc:mysql://localhost/sozluk";
            String username="root";
            String password = "";

            Connection con = DriverManager.getConnection(URL, username,
password);

            Statement stmt = con.createStatement();

            //String sql = "select * from kelime";
            ResultSet rs = stmt.executeQuery(sql);


            while(rs.next()){
                int flag = rs.getInt("flag");
                String word = rs.getString("word");
                //System.out.println(flag + " " + word);
                wordMap.put(word, flag);
            }

        } catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        return wordMap;


    }Map getWordsInDB2(String sql){
        Map wordMap = new HashMap();

        //Loar driver
        String driverName = "com.mysql.jdbc.Driver";
        try {
            Class.forName(driverName);

            //URL
            String URL = "jdbc:mysql://localhost/sozluk";
            String username="root";
            String password = "";

            Connection con = DriverManager.getConnection(URL, username,
password);

            Statement stmt = con.createStatement();

            //String sql = "select * from match";
            ResultSet rs = stmt.executeQuery(sql);


            while(rs.next()){
                String id = rs.getString("id");
                String path = rs.getString("path");
                //System.out.println(flag + " " + word);
                wordMap.put(id, path);
            }





        } catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
```

```
return wordMap;        }     }
```

Appendix E

- atılım üniversitesi nerededir?
- pamukkale üniversitesi bilgisayar mühendisliği bölümünün burslu kontenjanı kaçtır?
- atılım üniversitesi bilgisayar mühendisliği bölümünün 2007 yılında burslu taban puanı kaçtır?
- 2006 yılında atılım üniversitesi bilgisayar mühendisliği bölümü burslu taban puanı kaçtır?
- 2006 yılında atılım üniversitesi yazılım mühendisliği bölümü taban puanı kaçtır?
- atılım üniversitesi bilgisayar burslu puanı 2006 yılında kaçtır?
- atılım üniversitesi 2006 yılındaki bilgisayar mühendisliği burslu tavan kaç puandır?
- pamukkale üniversitesi bilgisayar mühendisliği burslu tavan puandır 2006 yılında kaçtır?
- pamukkale üniversitesi 2006 yılındaki bilgisayar mühendisliği burslu tavan kaç puandır?
- atılım üniversitesi bilgisayar mühendisliği bölümünde kaç öğretim üyesi bulunmaktadır?
- atılım üniversitesi bilgisayar mühendisliği bölümünde kaç öğrenci vardır?
- atılım üniversitesi yazılım mühendisliği bölümünün 2007 yılında kaç yayını vardır?
- pamukkale üniversitesi bilgisayar mühendisliği bölümünde öğretim üyesi sayısı kaçtır?