SIKÇA SORULAN SORULAR ÜZERİNDE ARAMA

Burcu YÜREKLİ
Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı
Kasım, 2008

JÜRİ VE ENSTİTÜ ONAYI

Burcu Yürekli'nin "***Sıkça Sorulan Sorular Üzerinde Arama***" başlıklı **Bilgisayar Mühendisliği** Anabilim Dalındaki, Yüksek Lisans Tezi 21.11.2008 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı-Soyadı	İmza
Üye (Tez Danışmanı):	**Yard. Doç. Dr. ÖZGÜR YILMAZEL**	**.....**
Üye	**: Doç. Dr. YUSUF OYSAL**	**.....**
Üye	**: Yard. Doç. Dr. HAKAN G. ŞENEL**	**.....**

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun …… tarih ve …… sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ABSTRACT

Master of Science Thesis

TURKISH FAQ RETRIEVAL SYSTEM

Burcu YÜREKLİ

Anadolu University
Graduate School of Sciences
Computer Engineering Program

Supervisor: Assist. Prof. Dr. Özgür YILMAZEL
2008, 43 pages

The number of accessible information resources via Internet is rapidly increasing. This rapid increase in information resources starts to be a problem in reaching appropriate data. Even though the search engines most of the time overcome this problem, for the Internet users it is a time consuming and frustrating process to search though the results and to find out the necessary data. Because of this, the need for systems that can quickly and correctly provide answers for user questions is raised.

The present thesis covers the performance evaluation of a Frequently Asked Question (FAQ) retrieval system that can retrieve the answer for a given natural language user question by matching it with the question answer pairs that are collected from FAQ web pages in Turkish. The aim is to retrieve the correct question answer pair as the first hit. To evaluate the performance, some of the questions from our data collection were selected and rephrased without changing the meaning, and a test collection is created. The effect of Turkish language specific analyzing and the effects of searching the user question in different fields are investigated. It is found that Turkish language specific analyzing improves the retrieval performance of the system. Also, including the title of the source to a search field increases the performance of the system.

Keywords: Information retrieval, FAQ, FAQ retrieval, Turkish

ÖZET

Yüksek Lisans Tezi

SIKÇA SORULAN SORULAR ÜZERİNDE ARAMA

Burcu YÜREKLİ

Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yard. Doç. Dr. Özgür YILMAZEL
2008, 43 sayfa

Günümüzde Internet aracılığıyla erişilebilen bilgi kaynaklarının sayısı giderek artmaktadır. Bilgi kaynaklarındaki hızlı artış bilgi erişim problemini getirmiştir. Arama motorları bilgi erişim problemini büyük çoğunlukla çözebilsede, Internet kullanıcıları için sonuçlar arasında gezinerek, ihtiyaç duydukları bilgiyi arayıp bulmak zaman ve istek kaybettirici bir işlemdir. Bu nedenle, kullanıcıların soracakları soruları hızlı ve doğru bir şekilde cevaplandırabilecek sistemlere duyulan ihtiyaç artmıştır.

Bu tezde kullanıcıların doğal dilde sordukları soruları web sayfalarından toplanan sıkça sorulan sorular ile karşılaştırıp, kullanıcının sorusunu cevaplayabilecek soru cevap çiftini ilk sonuç olarak getirmesi hedeflenmiş olan bir Sıkça Sorulan Soru Erişim Sisteminin performansı değerlendirilmiştir. Performansın değerlendirmesi için web sayfalarından toplanan sorulardan bazıları seçilip, farklı kişiler tarafından derlenerek test koleksiyonu oluşturulmuştur. Türkçeye özgü analizlerin ve farklı arama alanlarında yapılacak aramaların sistem performansına etkileri incelenmiştir. Türkçeye özgü analizlerin sistemin performansını artırdığı görülmüştür. Ayrıca, arama alanlarına soru kaynağına ait bilgilerin eklenmesi ile sistem performansının arttığı gözlenmiştir.

Anahtar Kelimeler: Bilgi Erişimi, Sıkça Sorulan Sorular, Sıkça Sorulan Soruların Çıkarımı, Türkçe

ACKNOWLEDGEMENTS

I would like to thank my advisor Asst. Prof. Dr. Özgür Yılmazel for his guidance and support during my study. It was my pleasure to work with him during this study.

Burcu Yürekli

November, 2008

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVATIONS

FAQ : Frequently Asked Question

IR : Information Retrieval

QA : Question Answering

TREC : Text Retrieval Conference

URL : Universal Resource Locator

WWW: World Wide Web

1. INTRODUCTION

Today, as Internet has become widely accessible, World Wide Web (WWW) becomes the most preferred information resource for many people. It carries huge amounts of online data and information services. This explosive increase of information turns out to be a problem. Knowing that the information we are looking for probably exists in somewhere in the Internet is not enough. People want to get appropriate information without spending a lot of time searching through the results retrieved by the search engines.

In the past decade, since finding and accessing to specific information became time-consuming, frustrating and labor-intensive for Internet users [1], Question Answering (QA) systems, which are developed to extract answers from collection of documents, as response to user questions posed in natural language, got important.

However, there are lots of duplicate and false information in the web and there is no guarantee among the answers of QA systems, so the design of these systems is one of the hardest fields in Computer Science. They require linguistic knowledge for answering correctly [2]. Even though a road map for QA systems is researched for a committee [3], open domain QA systems could not achieve the expected success.

Restricting the knowledge base of QA systems is one way to improve the accuracy of these systems, since there will be smaller number of documents [2]. Even though closed domain QA systems also have limitations and drawback, a QA system based on Frequently Asked Questions will have fewer problems. The system will not generate a new answer. Instead of generating a new answer for a given user question, QA system will match the existing questions with the user question. Also, since the question answer pairs in the knowledge base are mostly generated by experts, it will be guaranteed that, when a user question is correctly matched with an existing question, the answer will be correct.

In this thesis we construct a FAQ retrieval system on Turkish question answer pairs. FAQ retrieval system matches the given user question with the

question answer pairs in the data collection, and retrieves the most similar pair. To evaluate the performance of FAQ retrieval system different approaches were used. As the first step question answer pairs are fetched from several FAQ pages and a data collection is created. Since the evaluation of the system need to be done by questions that the exact answers are known, a number of questions were chosen from the data collection and rephrased. These rephrased questions were used as the test collection. We apply standard analyzes and Turkish language specific analyzes in retrieving the most similar question for each of the questions in the test collection. We compare the retrieval performance of our FAQ retrieval system, both for efficiency and effectiveness on several search fields and with different analyzers.

The organization of this thesis is as follows. Section 2 gives a background of the general concepts of QA and FAQ systems and briefly summarizes the related work done on FAQ systems. Section 3 presents the details of our FAQ retrieval study including the creation of the data collection and test collection, FAQ retrieval system setup, and retrieval experiments and obtained results with different indexing strategies and analyzers that were used in retrieving the most similar question answer pair as a response to a user question. Section 4 provides concluding remarks and future work.

2. BACKGROUND

2.1. Question Answering

2.1.1. Question Answering Systems

QA is a type of Information Retrieval (IR), which carries the process of search engines one step forward. The current search engines can return ranked lists of documents that may be relative to the information need; however, they cannot provide exact answers to the user. With the rapid increase of on-line data, need for automated question answering systems that retrieve answers to questions asked in natural language become important [4].

The task of QA system is for a given plain text question, to extract information from the system's knowledge base, which can be Internet web documents, databases or text documents, so to provide an intelligent answer back to the user [2]. To answer a question, first of all the QA system must understand the question written in natural language. Then the QA system must find out the answers from the knowledge base. In some appropriate form the system must present the extracted answers to the user with some supporting material, like a link to the original document that the answer extracted from [4]. This process can be seen as an understanding capability, which is difficult to implement and different from document retrieval.

2.1.2. Question Answering System Architecture

Several approaches have been developed for the design of automated QA systems. Even though for a specific system, the choice of components of the QA system can change; there are some general process elements that take place in some way for the implementation of these systems. Figure 2.1 shows the architecture of a QA system with these general elements.

Question Analyzer

The user's question, which is asked in natural language, is the input for the QA system. For the system to understand this question, it needs to be processed

and analyzed. It is the only input where the system can get information about what is being asked, so it should be processed to get as much information as possible.

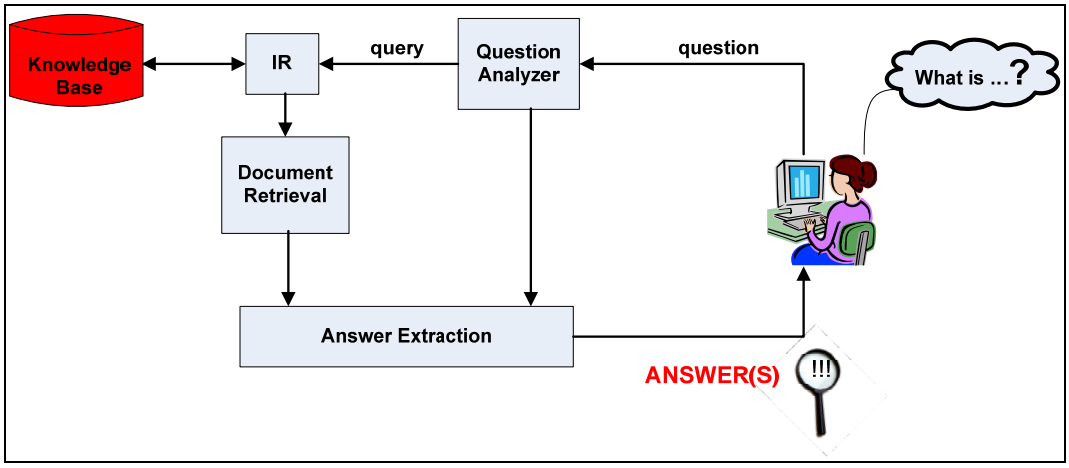

Figure 2.1. Question Answering Process

Information obtained in query processing is very important. This information will be used in candidate document selection that is the document retrieval process, so it affects the correctness of the answers.

With the question analyzer, the user's question is converted to a query string that will be submitted to the system. The terms in the query string depends on information that is extracted from the user question. The researchers implement various analyses for this process, some can be stated as syntactic parsing of the question, breaking down the question into tokens and eliminating the stop words, expanding the question terms with their synonyms, extracting the question type to identify the expected answer type, or expanding the query [2].

Document Retrieval

The query that has been obtained from the question analyzer will be applied to the knowledge base to select candidate documents. Candidate documents are the ones most likely to contain the answer of the user question.

Depending on the knowledge base, search can be done on several ways. If the knowledge base is web documents, custom search engines or third party search engines can be used to search the web to extract the relevant documents. On the other hand, search can be done on structured or unstructured data stored on a database. As the search method, Boolean keyword matching or term weighting can be used. Some researchers think that preprocessing of the collection increase the performance of the QA system. Techniques like sentence splitting, part of speech tagging, or named entity annotation, can be applied to the collection, before searching the entire knowledge base [2].

Document retrieval process of the QA systems is very similar to the retrieval process of the IR systems. However, their aims are different; in QA systems finding the answer for a given question is the key point.

Answer Extraction

In this process, the query representation of the user question and the candidate documents obtained in the document retrieval process are matched against each other, and a set of candidate answers are produced. These answers are ranked in the means of correctness and the most relevant answers are displayed to the user in some format [4].

Answer extraction is not a simple task. The steps followed in this approach changes from system to system depending on the methods applied in the question processing and document preprocessing. The factors that will effect answer extraction can be the complexity of the user question, the answer type provided by question processing, the search method, and the context or the focus of the question.[1] To rank the passages that are extracted as the possible candidates containing the answer, Information Extraction algorithms are used.

QA systems use different methods to show the extracted answer to the user. Some systems respond with the passage that contains the answer, others try

---

[1] http://en.wikipedia.org/wiki/Question_answering

to formulate the answer or answers by extracting proper parts from the retrieved documents; this is especially when the answer is spread out different documents.

2.1.3. Open-domain QA Systems versus Closed-domain QA Systems

The knowledge base of the QA systems can vary from small number of local documents to whole WWW. Depending on the content of the knowledge base, these systems are either called closed-domain QA systems or open-domain QA systems, respectively.

Open-domain QA systems are capable of extracting the answer of the user query from unrestricted domain of documents. By the evolution of the Internet, huge amounts of indexed documents become easily accessible through search engines. The idea of finding information need without spending a lot of time searching through the results of the search engines, turns the focus on open-domain QA systems. Also, TREC-QA[2] track competitions introduced organizations and universities to open domain QA research.

Open domain QA systems are successful to answer a factoid question. Since, there are lots of available data on the Internet, when the search engine either queries the exact question, or queries by rewriting the user question as an answer, probability of obtaining a sufficient answer is very high. Even though the performance of these systems is sufficient on factoid questions, they cannot provide answers to more complex questions, like comparison questions. Containing such large document collection has also some drawbacks for open domain QA systems. In the document retrieval process if the user query is too general, too many documents may be retrieved. If it is too specific the retrieved documents may not contain the proper answer. Moreover, it is impossible to control the quality of the content on large document collection. There can be misspelling or badly formatted text or inconsistent and redundant information. These factors will conflict the extracted answers, especially for changing data [2].

---

[2] http://trec.nist.gov/

On the other hand, closed-domain QA systems provide answers that are about a specific domain, such as medicine or student advising, so they have a restricted domain of knowledge.

Closed-domain QA systems have some conveniences that are not possible to obtain in open-domain QA systems. Closed domains can have many technical terms that play an important role in determining the answer of a question, which will be discarded as misspelled words in open-domain QA systems. The words, which can have several different meanings, will have a restricted meaning on closed domains, so the complexity of question analyzing will be limited. Also, due to the limited number of domain specific words in closed-domain QA systems, creating a lexicon, dictionary or ontology is easier than open domains. More importantly, since the knowledge base is limited in closed-domains, the answers are qualified and reliable. Closed-domain QA systems also have some disadvantages. Due to the scarcity of data, it is difficult to write Information Extraction algorithms that extracts information related to the user query and discards the rest of the passage in the final answer. In other words, it is hard to analyze matching passages that can increase the accuracy of the answer. Generating answers to complex questions, like why and how type questions is a major problem for both open-domain and close-domain QA systems, since these kinds of questions require more in depth answers. Open-domain QA systems have several metrics for performance evaluation, but these metrics are limited for closed-domain QA systems, so it is challenging to accurately evaluate them. Based on the target user different evaluation techniques are required [2].

2.1.4. QA Systems versus IR Systems

Information retrieval can be defined as:

"…finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)" [5].

The aim in IR is that in response to a user query, retrieve most relevant documents among a large document collection, which can be any type of source

where text can be extracted. Search engines are the most visible IR applications. With the increase in online data, search engines become crucial in information access. The architecture of IR systems can be described in three main parts as Indexing, Searching and Ranking, which is shown in Figure 2.2.

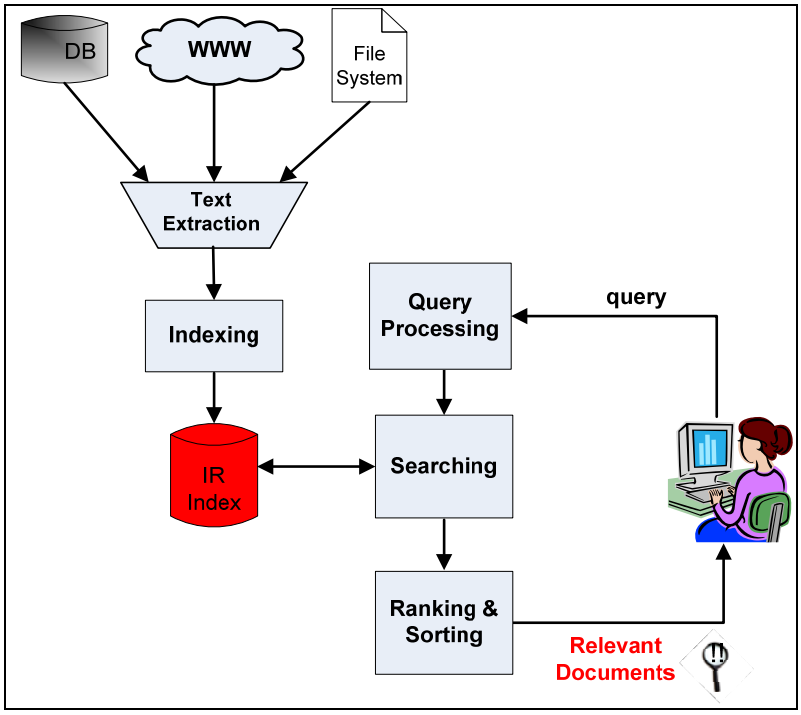

Figure 2.2. Information Retrieval Process

IR systems and QA systems are different in design and process. These two systems have different purposes.

IR system retrieves relevant documents in response to user query, and users left to extract answers by themselves. QA systems try to extract answer in response to user questions.

Also, they have different inputs. QA systems get a natural language questions from the user, IR systems get Boolean queries.

QA systems deliver one or more exact answers and their context, IR systems return snippets and links to the documents that snippet extracted from.

"Think of IR as finding right book in a library".

"Think of QA as a librarian giving you the book and opening it on the page with the information you're looking for "[3].

In IR systems the user can decide whether the documents returned as relevant to a query really answer the query or not. The user knows what he is looking for and can judge the returned answers. However, in QA systems user expects the system to answer a question that he does not know the answer. So, since the user does not have an idea about the answer, he cannot judges the responded results.

2.2. FAQ Answering

2.2.1. FAQ

"One popular way of sharing inquiry based information on the Internet is by listing frequently asked questions (FAQ) and their answers on web pages or in text documents. " [2].

In many web sites, there exist a section namely, "Frequently Asked Questions" or FAQ, which contains certain questions and their answers, which come up over and over or which supplier sees necessary to answer in advance. FAQ become so popular that, today we can find FAQ popping up just about anywhere. Call-center manuals are an example of FAQ file. Also, FAQ files take place in many forums and web portals to store knowledge on the site users' questions [6].

The idea behind a FAQ file is to record an answer about some common question and make that question answer pair available, particularly to newcomers who may otherwise ask the same question again [7].

---

[3] http://twiki.di.uniroma1.it/pub/Estrinfo/Materiale/QA-1.ppt

Before the Internet, the call-centers are the only places where users can get answers to their questions about a service or a product. As WWW has become widely accessible, FAQ turn out to be the main method to supply and get help online. Many Internet based companies and organizations direct their users to publicly available FAQ, before asking question to support personnel. Looking at the companies and organizations side, FAQ restrict the amount of redundant questions and reduce the resources spent on support personnel [2]. The users also get benefit, by the help of FAQ. They get answers to their specific questions without searching the web or calling the call-center.

The origins of the FAQ are based on ARPAnet[4] mailing lists. Later they became available on UseNet, which is the most common place to find FAQ files, in the form of "*.answers" moderated newsgroups [2].

USENET is a worldwide distributed Internet discussion system[5]. It is constructed by individuals and small groups organized around their particular interests and hobbies, which users can read and post public messages to newsgroups. Specific to the subject of the USENET newsgroup, many of them have a FAQ section [8]. The Internet FAQ Archives web site[6], which has more than 4000 FAQ archived, contains USENET FAQ postings. It is a repository of online FAQ files and a major knowledge base about almost any topic, which is exponentially growing everyday. It allows users to find FAQ files using keywords. There are several more web sites that are building as an online FAQ repository. For example, The Dynamic FAQ Database[7] is a web application that provides its users to construct FAQ databases about a specific web site. The

---

[4] http://en.wikipedia.org/wiki/ARPANET

[5] http://en.wikipedia.org/wiki/USENET

[6] http://www.faqs.org

[7] http://products.dynamicwebdevelopers.com

Dynamic FAQ Database allows users to search through these FAQ databases and submit questions [9].

2.2.2. FAQ Answering System

FAQ can be found all over the Internet and users can access them through web sites or FAQ repositories. They are generally very useful, especially they work well for people who are interested on a certain subject and who know where to look for an answer [7]. On the other hand, they can also be troublesome.

When there is a question, few people know where FAQ files can be found. Even if they know where, because a collection of FAQ file is generally organized as a simple list, few of them know which FAQ files or which categories to look for. This means, even though information is available in theory; it is hidden from the more occasional user [10].

Eriks Sneiders states several disadvantages of organizing FAQ files as lists in his paper [11]. He states them as follows:

Since the user does not have a chance to ask a question, he has to scan through the list of existing questions or several lists of various questions one by one to find the best question which answers his own question. However, if the set of FAQ list is too long, it is very difficult and time-consuming to scan the entire questions, especially for the ones who are not interested on learning every detail. In this case, the user can try a keyword match among this large text mess, but this is also impossible if the list is spread over several documents or pages. Even though semantic grouping of the FAQ may be helpful, if there are overlaps between the groups, it may be ambiguous. So, this grouping should be done correctly. Also, the user may not know exactly which group to look for finding the question. Having several FAQ in the list that answer the user's question may be another problem, if the list is not well structured. In this case, the user should scan through the list in order not to loose valuable information. These situations often force the user to give up [11].

In some systems keyword search capability is given to users. However, for specific questions, this search results cause the user to go through the results and

analyze them. Also, to get qualified results, the user should know what keywords to use in the search. Finding answers to uncommon questions is harder by this method. Moreover, the returned results may not contain the correct answers or they may be wrong [2].

Search engines are also not successful in finding an answer for a given natural language question, since they are not designed for this aim. A research done by Radev, Libner and Fan [12] shows that when a complete natural language question is searched by a search engine, correct answers are found 75 percent of the time somewhere in the top 40 documents.

To provide more precise answers instead of a list of links, there is a need for FAQ answering systems, which extend the aim of FAQ repositories. FAQ answering systems are designed to retrieve FAQ upon a user request expressed in natural language. In other words, as a response to an incoming user question these systems try to match it with the question answer pairs that take place in FAQ repository. If the user question is similar to one of them, system returns that pair as an answer.

2.2.3. Related Work on FAQ Answering Systems

In the literature, most of the researches have been done on open-domain QA systems. There is not much work done by using FAQ and their answers as a knowledge base.

FAQFinder is one of the most famous QA systems that is based on FAQ retrieval. This system is developed on UseNet newsgroups' frequently asked question answer pairs. It is designed by combining three technologies which are statistical IR, syntactic parsing and semantic concept matching. In the IR part, START is used to retrieve the relevant FAQ for the user question. With a syntactic parser FAQ Finder identifies verbs and nouns in a user query. Then by using WORDNET as semantic knowledge; system selects possible matches between the words in the user query and the words in FAQ pairs [10].

Auto-FAQ is another QA system that is based on FAQ retrieval. As in FAQFinder, this system is also based on UseNet newsgroups' FAQ pairs.

However, the difference is that in Auto-FAQ these FAQ pairs are locally maintained and there is no search on external resources. In Auto-FAQ, keyword comparison method based on natural language processing techniques was used to match user query to FAQ pairs [13].

Sneiders Automated FAQ Answering system also uses FAQs. This system is based on an algorithm that he named as Prioritized Keyword Matching. He classified the words into three types as required keywords, optional keywords, and irrelevant keywords. The required keywords are the primary words that should exist and cannot be ignored, the optional keywords are the words that help in the meaning, but do not have to exist. The irrelevant words are the common words, which are called as stop words in IR. Also, there are forbidden word, whose existence is not compatible. Before user queries were submitted to the system, each FAQ in the database were analyzed by an expert. For each FAQ, the expert determines these keywords and enters them manually. By considering these keywords system retrieves and ranks the most relevant FAQ for a given user question [11].

Later, to improve the retrieval results, cluster-based retrieval techniques [14] and ontology based retrieval techniques [15] were applied on FAQ based systems.

3. FAQ RETRIEVAL STUDY

In this thesis we construct a FAQ retrieval system based on question answer pairs collected from FAQ web pages written in Turkish. For each question in the data collection, there exists only one answer and our aim is to retrieve the exact answer for a user question as the first hit.

Our FAQ retrieval system includes three main parts as creation of data collection by collecting question answer pairs from FAQ web pages, creation of test collection to use in evaluation of the system and retrieving the similar questions.

Each of these parts will be described in details. The order is like this. In subsection 3.1 the creation of the data collection and in subsection 3.2 the creation of the test collection will be described. In subsection 3.3 we will describe the steps in system setup. FAQ Retrieval Experiment and corresponding results will be given in subsection 3.4.

3.1. Data Collection

3.1.1. Fetching FAQ Pages from the Web

To create a data collection, the first step was to find out the web pages that contain the lists of question answer pairs written in Turkish. To find these FAQ pages we could either use a web crawler or a search engine. However, the number of the FAQ pages, especially written in Turkish, is extremely small compared to whole web, so collecting these pages using crawling would be resource inefficient. Instead, Google[8] is used to locate these FAQ pages that may contain FAQ.

Most of the FAQ pages written in English, either has the word “faq” in the title or in the URL of the web page, a query like “intitle:faq” or “inurl:faq” can be asked to Google. Even though many potential FAQ pages can be missed, the

[8] http://www.google.com.tr/

number of responded pages is extremely large [16]. We try these queries to find FAQ pages in Turkish. Since Turkish translation of "Frequently Asked Questions" is "Sıkça Sorulan Sorular", instead of the string "faq", we used the string "sss", which is the acronym for "Sıkça Sorulan Sorular". When a query like "intitle:sss" among web pages in Turkish is asked, Google reports **128.000** hits, and for a query like "inurl:sss" among web pages in Turkish is asked, Google reports **399.000** hits. Unfortunately, among the hits most search engines provide, at most 1000 hits are relevant for any given query; so to obtain more ideal hits we expand these queries. After researching several FAQ pages in Turkish, we conclude with some observations:

- The URLs' of the FAQ pages in Turkish are not always contain the string "sss".
- Instead of "Sıkça Sorulan Sorular", in some FAQ pages people used "Sık Sorulan Sorular", "Çok Sorulan Sorular", "En Çok Sorulan Sorular" or "Sık Sık Sorulan Sorular" as a title.

These alternative phrases are also used in our searches to expand the search region.

Among the hits returned by Google, we manually chose **155** FAQ pages written in Turkish. The main reason of manual choosing is that, not all the questions that exist in a FAQ page were written as a question answer pair. Some of them are written as descriptions or explanations of subjects.

3.1.2. Extracting Question Answer Pairs

Once we decided on the FAQ pages that the data collection would be created from, we automatically extracted the question answer pairs from these FAQ pages and stored them in database. We wrote a Java program to automatically extract these question answer pairs.

This program gets the URL of the FAQ page as an input. For a given URL, if the answers of the questions are on the same page with the questions, program parses these question answer pairs. Then, each extracted question answer pair and the URL of the page are stored into database. On the other hand, if the given page

contains only the questions, but the answers of the questions are given on separate pages with hyperlinks; to extract the required answer, program goes to the page the question directs. Then, it gathers the question answer pair from the directed page and stores the pair into database with the new URL.

The main difficulty of this process is the layout differences between FAQ pages. This difference can also be seen visually by glancing.

The layouts of the FAQ pages vary depending on their design. Special HTML tags give us a clue about how question answer pairs on the web page can be parsed. These tags can either be a table-row (tr), a div with a special css class, an ordered or an unordered lists (ul, ol) or header tags (hX). Even though the HTML markup tags that surround the questions and answers are different depending on the FAQ page, they were helpful in detecting the questions and their answers. Question answer pairs are either marked as separate paragraphs, or the questions are marked by using italic, bold, different size of fonts, different colors or css classes. Several prefixes can also be used to mark them.

Our observations showed that on Turkish FAQ pages, questions were commonly prefixed with things like:

- S: or S. or S
- Soru: or Soru
- 1: or 1. or 1) or 1
- Q: or Q.

In some Turkish FAQ pages, answers were also prefixed with things like:

- C: or C. or C
- Cevap: or Cevap
- A: or A.

We limited the HTML text blocks by finding the beginning and the ending of the section that contains the question answer pairs. Then by using either question prefixes or HTML styles, we extracted the questions. The positions of

the answers helped us in extracting the answers. For some of the FAQ web pages that contain answer prefixes, we used these prefixes to extract the answers of the corresponding questions.

As in many of the web pages, some of the FAQ pages which we choose were not well formatted. There were a lot of non-nested or miss closing tags. To overcome these we used simple heuristics.

As stated before, our program inserted these question answer pairs into database that was used only to store the data. For each question answer pair, we stored question, answer, URL and source as attributes. This source attribute is related with the table that contains information like, the title, and the URL of the source FAQ page, which the question answer pair was extracted. Since the title of a FAQ page gives information about the content of its question answer pairs, it was important in our study and we used that information in our experiments.

To control the duplicates, while inserting a question answer pair into database we checked whether the question is different from any existing one. Since duplicate questions decrease the performance of the retrieval, if a question already existed in database, this question answer pair was ignored.

As a result of this fetching part, we extracted 2762 question answer pairs from 155 different FAQ pages. These question answer pairs constructed our data collection that our FAQ retrieval system was based on.

3.2. Creation of Test Collection

3.2.1. Selection of Questions

To test the performance of our FAQ retrieval system, we created a test collection. Among 2762 question answer pairs we randomly chose 30 questions of various types. The distribution of these 30 questions is shown in Table 3.1.

Table 3.1. Distribution of Question Types in Test Collection

Question Type	Count
How	*13*
What	*3*
When	*1*
Where	*2*
Which	*3*
Why	*2*
Yes/No	*6*
Total	**30**

3.2.2. Rephrasing the Selected Questions

The selected questions were given to 20 different participants and they were asked to rephrase these questions. With out showing the answers of the questions, we asked them to rewrite these 30 questions in different forms with out changing the meaning. Then we gathered the rephrased questions from our participants. Totally we had 205 rephrased questions which we have used as a test collection. We gained at least 5 different rephrased questions for each original question in the test collection. As an example one of the selected question and its rephrased forms are shown in Table 3.2.

3.2.3. Quality of the Rephrased Questions

Since test collection is extremely important in the performance evaluation of the FAQ retrieval system, all the questions returned by the participants were manually checked for errors. For eliminating rephrasing errors we ignored some of the rephrased questions, which were duplicated and which were rephrased incorrectly in meaning.

Table 3.2. Example of a Question and its Rephrased Forms

Original Question: *"Emlak vergisine ait bildirimi süresinde vermeyen mükellef adına hangi cezalar kesilir?"*
Rephrased Questions: • *Emlak vergisi bildirimini zamanında vermeyen kişilere uygulanan cezalar nelerdir?* • *Emlak vergisine ait bildirimi zamanında teslim etmeyen mükellef hangi cezalara çarptırılır?* • *Emlak vergisine ait bildirimi süresinde vermemenin cezaları nedir?* • *Emlak vergisi bildirimini zamanında yapmamak hangi cezai yaptırımları gerektirir?* • *Emlak vergisi bildirimi süresinin aşımı halinde uygulanacak cezalar nelerdir?* • *Emlak vergisi bildirimini süresinde yapmayan kişilere hangi cezalar uygulanır?* • *Emlak vergisi bildirimini süresinde yerine getirmemenin cezası nedir?* • *Emlak vergisi bildirimini zamanında yapmamanın cezası nedir?* • *Mükellef emlak vergisi bildirimi yapmazsa ne ceza kesilir?*

3.3. Retrieval System Setup

The aim of our FAQ retrieval system is to choose the most similar question with the entered user question from the data collection. We used Apache Lucene[9], which is a high performance IR library written in Java, as the core of our FAQ retrieval system.

3.3.1. Indexing

The data collection is indexed to obtain the format that can be comprehended by Lucene. The most used data structure for this is inverted index. Inverted index is like a dictionary that contains all the unique terms, takes place in the data collection. The aim of inverted index is to avoid linear searching and to give fast search capability. For each term in the dictionary, information about the documents containing this term and the position of the term in that document is recorded as a posting list.

3.3.2. Searching and Ranking

To retrieve the question which satisfies the user question best, the similarities between every question in data collection and the user question should be calculated. In order to display the most similar question, the result set should be ranked according to relevancy. The vector space model is the most widely used method in IR to rank the result set. This model lies on the bases of the similarity measures used by most web search engines[10].

In vector space model each document in the collection, which is a question answer pair for our system, and a given user query is represented as a vector. The dimension of each vector is same as the number of unique terms in the document collection. Each unique term corresponds to a dimension and represented with an index term weight, which will be used in computation of similarity between

---

[9] http://lucene.apache.org

[10] http://www2002.org/CDROM/refereed/643/node5.html

documents and queries. If a unique term does not exist in a document, then the term weight value of that unique term will be zero.

Let's say that there are n documents in the data collection, and the number of unique terms is m. Then the dimension of a vector will be m. The n^{th} document will be represented as

$$d_n = \left[weight_{1,n}, weight_{2,n}, ..., weight_{m,n}\right]^T$$

The user query will be represented as

$$q = \left[weight_{1,q}, weight_{2,q}, ..., weight_{m,q}\right]^T$$

The similarity in vector space model is determined by the inner product of the document vector and the query vector, in which the word overlaps indicate similarity. Even though there are other measures, cosine similarity is the most popular similarity measure. Cosine similarity measures the angle between the document vector and the query vector. The equation of cosine similarity is given in Equation (2.1), where d_i is the document vector and q is the query vector.

$$sim(d_i, q) = \frac{d_i \bullet q}{\|d_i\| \times \|q\|} = \frac{\sum_{k=1}^{m} weight_{k,i} \times weight_{k,q}}{\sqrt{\sum_{k=1}^{m} weight_{k,i}^2} \times \sqrt{\sum_{k=1}^{m} weight_{k,q}^2}} \qquad (2.1)$$

Since the numerator is the dot product and the denominator is the product of Euclidean lengths of document and query vectors, this measure gives the cosine of the angle between q and d_i vectors. This equality is given in Equation (2.2). If the cosine of this angle is equal to zero, then this means there is no match between q and d_i vectors.

$$sim(d_i, q) = \cos ine\,\theta_{d_i} \qquad (2.2)$$

In calculation of term weights ($weight_{k,i}$), there exists different approaches. The term frequency-inverse document frequency ($tfidf$) weighting is

the best known method in IR. To produce a composite weight for each term in each document, term frequency ($tf_{t,d}$) is multiplied by inverse document frequency (idf_t) as given in Equation (2.3).

$$weight_{t,d} \ \ with\ tfidf = tf_{t,d} \times idf_t \tag{2.3}$$

$tf_{t,d}$ - denotes the number of occurrences of term *t* in document *d*.

idf_t - denotes the inverse of document frequency and obtained by dividing the number of all documents (*n*) by the number of documents that contain the term t (which is the document frequency for term *t,* and shown as df_t), and then taking the logarithm of this division.

$$id_f = \log \frac{n}{df_t} \tag{2.4}$$

When Equation (2.4) substituted in Equation (2.3), the weight value of term *t* in document *d,* can be given as in Equation (2.5).

$$w_{t,d} = tf_{t,d} \times \log \frac{N}{df_t} \tag{2.5}$$

3.3.3. Performance Evaluation Methods

Precision – Recall

Precision and recall are two measures which are commonly used in evaluation of the quality of IR systems. These measures depend on results of a query and the relation between these results and all relevant and non-relevant documents.

The ratio of "*the number of relevant documents retrieved by a query*" to "*the total number of documents retrieved by a query*" gives the precision.

$$\text{Precision} = \frac{\text{\# of relevant document retrieved}}{\text{total \# of documents retrieved}}$$

Recall is the ratio of *"the number of relevant documents retrieved by a query"* to *"the total number of existing relevant documents, which should have been retrieved"*.

$$\text{Recall} = \frac{\text{\# of relevant document retrieved}}{\text{total \# of existing relevant documents}}$$

Mean Reciprocal Rank (MRR)

MRR is one of the evaluation methods that are used in TREC QA track to evaluate the performance of a system.

"Mean **reciprocal rank** is a statistic for evaluating any process that produces a list of possible responses to a query, ordered by probability of correctness"[11].

The reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer. The average of the reciprocal ranks of results gives the mean reciprocal rank for all the queries. The equation of MRR is shown in Equation 2.6.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{Q}\frac{1}{rank_i} \tag{2.6}$$

In our system, since there is only one correct answer for each question, if the correct result is responded as the first answer, then it will get 1; or else if it is

[11] http://en.wikipedia.org/wiki/Mean_reciprocal_rank

responded as the second answer, then it will get 1/2. Depending on the chosen rank level, this value can go up to 1/rank. The average of these values given for each query will be the MRR of the system.

3.3.4. System Overview

The FAQ retrieval system is summarizes in Figure 3.1.

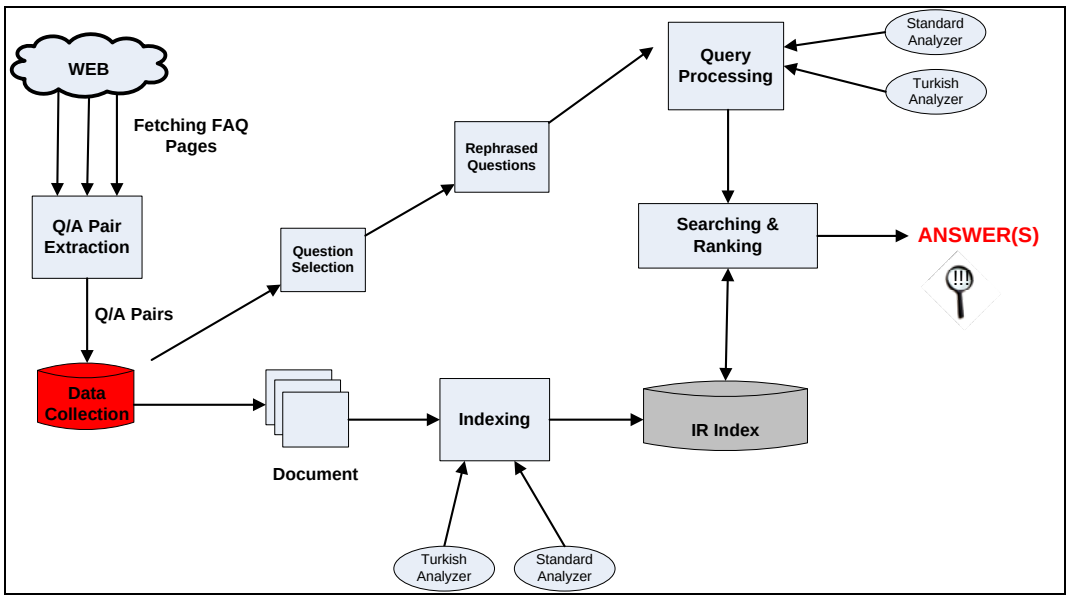

Figure 3.1. System Overview

3.4. FAQ Retrieval Experiments

To understand the effectiveness of our FAQ retrieval system's search capability on question answer pairs, we report on two groups of experiments. In both of the experiments we used the data collection which contains 2762 question answer pairs in Turkish that have been collected from 155 different FAQ pages.

To evaluate the performance, rephrased questions were given to system as a query. For each of the 205 rephrased questions, the retrieved question answer pairs were examined. For us, the most important criterion was whether the correct question answer pair retrieved as the first hit or not. Since, the aim in developing the FAQ retrieval system is to retrieve the exact answers for a given user question. In our experiments we tried to improve the performance of the system in this way.

Our experiments are based on two different analyzers. Each of the analyzers was used in indexing of the data collection and query parsing. One of these analyzers is the Lucene's Standard Analyzer, which indexes all the words based on JavaCC-based grammer and recognizes alphanumerics, numbers, e-mail addresses [17]. The other analyzer is a Turkish language specific analyzer, which performs stemming on words to remove inflectional suffixes, stop-word elimination and synonym filtering [18].

3.4.1. Experiments and Results with Question Indexing

In the first group of experiments, each question answer pair that is stored in database, written in individual XML files. These XML files consist of seven fields, namely id, url, question, answer, sourceId, sourceTitle, sourceUrl. An example of a FAQ document is shown in Figure 3.2. Among these seven fields question, answer and sourceTitle fields contain searchable textual information. To compare the retrieval results, we constructed some more fields by simply concatenating different combinations of these fields. Then these fields were used in indexing.

```
- <doc>
    <id>1002</id>
    <url>http://www.thy.com/tr-
      TR/help/faq/miles_smiles/online_award_ticket.aspx</url>
    <question>Online ödül bilet rezervasyonu yaparken başka
      birisinin kredi kartını kullanabilir miyim?</question>
    <answer>Kredi kartı sahibi rezervasyonunuzdaki yolculardan
      biri olmalıdır.</answer>
    <sourceId>41</sourceId>
    <sourceTitle>THY Online Ödül Bilet</sourceTitle>
    <sourceUrl>http://www.thy.com/tr-
      TR/help/faq/miles_smiles/online_award_ticket.aspx</sourceUrl>
  </doc>
```

Figure 3.2. FAQ Document

For each question answer pair we index the following fields:

- ***question:*** only the question is indexed
- ***answer:*** only the answer is indexed

- ***question + answer:*** a new field is created by concatenating the question and the answer part, and that field is named as ***qa***.

- ***sourceTitle + question:*** a new field is created by concatenating the title of the source and the question part, and that field is named as ***srctitleq***.

- ***sourceTitle + question + answer:*** a new field is created by concatenating the title of the source, the question and the answer, and that field is named as ***srctitleqa***.

In our runs we used *id* as a unique identifier and *question*, *answer*, *qa*, *srctitleq*, and *srctitleqa* as textual fields.

Since there were 2762 question answer pairs in our data collection, all of these question answer pairs took place in our index. To compare the performance of our system depending on search fields, each of the queries were searched separately on *question*, *answer*, *qa*, *srctitleq* and *srctitleqa* fields. Table 3.3 shows the performance of our system based on Standard Analyzer and Table 3.4 shows the performance of our system based on Turkish Analyzer on these search fields. The results obtained by retrieving the first 10, 5 and 3 results were also given in the tables.

These results showed us that when a query is searched on *srctitleq* field for both the analysis based on Standard Analyzer and Turkish Analyzer; the retrieved results were more relevant than the results obtained by searching on other fields. The number of relevant documents retrieved and the number of relevant documents that retrieved as the first hit by the system had the highest scores by searching on *srctitleq* field. s*rctitleq* field contains the title of the source and the question.

For example:

Question*:* Etkinleştirme nedir?

Source: Orijinal Microsoft Yazılımları

Srctitleq Field: Orijinal Microsoft Yazılımları Etkinleştirme nedir?

Table 3.3. Results based on Standard Analyzer on Question Answer Index

Search Field \ Results		*Among First 10 Results*		*Among First 5 Results*		*Among First 3 Results*	
		%	**#**	**%**	**#**	**%**	**#**
question	*Relevant*	92.68	190	90.73	186	89.75	184
	1^{th} *Hit*	86.82	178	86.82	178	86.82	178
answer	*Relevant*	62.43	128	46.34	95	39.02	80
	1^{th} *Hit*	23.9	49	23.9	49	23.9	49
qa	*Relevant*	90.73	186	85.85	176	84.39	173
	1^{th} *Hit*	76.58	157	76.58	157	76.58	157
srctitleq	*Relevant*	**97.07**	**199**	**94.14**	**193**	**93.65**	**192**
	1^{th} *Hit*	**88.29**	**181**	**88.29**	**181**	**88.29**	**181**
srctitleqa	*Relevant*	91.21	187	86.34	177	83.41	171
	1^{th} *Hit*	76.09	156	76.09	156	76.09	156

As can be seen from the example, some questions may be so general and related to more than one subject. For these short questions it is hard for the system to retrieve the correct result, since there can be some other questions in the data collection similar to the asked one, but about a different subject. Concatenating the title of the source with the question, as in the *srctitleq* field, and searching on this field helps the system to overcome the ambiguity about the subject of the asked question.

Table 3.4. Results based on Turkish Analyzer on Question Answer Index

Search Field \ Results		*Among First 10 Results*		*Among First 5 Results*		*Among First 3 Results*	
		%	**#**	**%**	**#**	**%**	**#**
question	*Relevant*	98.53	202	97.56	200	97.07	199
	1^{th} *Hit*	91.7	188	91.7	188	91.7	188
answer	*Relevant*	74.14	152	58.53	120	47.31	97
	1^{th} *Hit*	30.73	63	30.73	63	30.73	63
qa	*Relevant*	94.14	193	91.21	187	88.29	181
	1^{th} *Hit*	80.48	165	80.48	165	80.48	165
srctitleq	*Relevant*	**99.51**	**204**	**98.04**	**201**	**96.09**	**197**
	1^{th} *Hit*	**94.14**	**193**	**94.14**	**193**	**94.14**	**193**
srctitleqa	*Relevant*	94.63	194	92.68	190	89.26	183
	1^{th} *Hit*	80.48	165	80.48	165	80.48	165

Another observation on these results was that our system retrieved more relevant results with Turkish Analyzer than the Standard Analyzer. For 205 rephrased questions, 181 of them retrieved correctly as the first result by our system with Standard Analyzer. On the other hand, with Turkish Analyzer 193 of the rephrased questions retrieved correctly as the first hit. This showed that stemming, stop word elimination and synonyms are necessary pre-processing methods to increase retrieval efficiency on Turkish.

3.4.2. Experiments and Results with Page Indexing

We thought that first matching the user question with a source took place in our data collection, and then searching the relevant question answer pair among the questions that were in this source, could increase the performance of our system. To apply this idea, we created two different indexes. One index for the source, which was called as Page Index and the other index for question answer pairs that was called as Question Index.

Page Index: For each source stored in database, we created individual XML files that contain three fields, namely *sourceId*, *sourceTitle* and *sourceText. sourceText* field is generated by concatenating all the question answer pairs taken from that source or FAQ web page. An example of a page document is shown in Figure 3.3. In our runs we used *sourceId* as a unique identifier and *sourceTitle* and *sourceText* as textual fields.

Since in our data collection there were 155 FAQ pages, these 155 sources took place in our Page Index.

```
<doc>
  <sourceId>78</sourceId>
  <sourceTitle>Boğaziçi Üniversitesi Superdorm</sourceTitle>
  <sourceText>Başvuru için gerekli belgeler neler? Superdorm'da kalmakta olan öğrenciler, Başvuru
    Formu doldurmak, 1 adet yeni çekilmiş fotoğraf, İkametgah Belgesi ve ilk taksitin yattığını
    gösterir Dekont getirmek zorundadırlar. Superdorm'a ilk defa başvuracak öğrenciler ise; Başvuru
    Formu (Eksiksiz ve okunaklı doldurulmuş), İkametgah, Sabıka kaydı, Vukuatlı nüfus kayıt örneği,
    2 adet yeni çekilmiş fotoğraf, Dekont (1. Depozito 2. Başvuru taksiti) ile kayıt olabilirler.
    . . .
    . . .
    şeyler asılamaz. Yaz tatilinde İstanbul'a gelirsem Superdorm'da kalabilir miyim? Yaz
    döneminde, ancak Yaz Okulu'na devam edenler Superdorm'da kendilerine ayrılan blokta olmak
    kaydıyla konaklama yapabilirler. Dönem ortasında Superdorm'dan ayrılırsam para iadesi var mı?
    Dönem ortasında Superdorm'dan ayrılınması halinde para iadesi yapılmaz. Ayrıntı için
    Sözleşme'nin ödemelere ilişkin maddelerini inceleyiniz.</sourceText>
</doc>
```

Figure 3.3. Page Document

Question Index: The Question Index was same as the index described in the first experiment, which contains the question answer pairs in the data collection.

The user question first searched on page index. With this search the source ids' of the most relevant sources (the number of the retrieved sources depended on the user) for the given question was found. Then the user question is searched on the question index, but among the question answer pairs whose source id were one of the source ids obtained from the search on page index. Also for this experiment both Standard Analyzer and Turkish Analyzer were used and compared. Table 3.5 shows the performance of our system based on Standard Analyzer and Turkish Analyzer for experiment two. These results are based on the first 3 relevant sources retrieved from the page index. We also tried this experiment for different number of sources, but the best results were obtained by number of 3 relevant sources. As in the first experiment, searching on *srctitleq* search field provided the highest performance, because of this only the results on this field were shown.

When the retrieval results obtained from first group of experiments compared with the retrieval results obtained from second group of experiments, it could be stated that the retrieval efficiency of the system had increased. Based on standard analyzer, the number of correctly retrieved first hits was 181 in experiment one, but now it is 186. Also the number of correctly retrieved first hits of the system based on Turkish analyzer increased from 193 to 194, which was important for a small data collection like ours.

Table 3.5. Comparison of the results based on Standard Analyzer and Turkish Analyzer, where the first 3 relevant sources retrieved from the page index and then used on question index on *srctitleq* search field

Search Field \ **Results**		***Among First 10 Results***		***Among First 5 Results***		***Among First 3 Results***	
		%	**#**	**%**	**#**	**%**	**#**
Standard Analyzer	*Relevant*	97.56	200	96.09	197	95.12	195
	1^{th} *Hit*	90.73	186	90.73	186	90.73	186
Turkish Analyzer	*Relevant*	99.51	204	98.04	201	97.56	200
	1^{th} *Hit*	94.63	194	94.63	194	94.63	194

4. CONCLUSIONS AND FUTURE WORK

This thesis covers evaluation of a FAQ retrieval system that retrieves answers to user's questions from frequently asked question answer pairs collected from Turkish FAQ pages on the web. Creation of the data collection by fetching question answer pairs from web, creation of the test collection and retrieving similar question answer pairs corresponding to the created test collection were the main steps in development of our FAQ retrieval system.

The test collection was created by rephrasing the selected questions from the data collection. By knowing the relevant question answer pair that should be retrieved by the system for each of the rephrased questions in the test collection, we could evaluate the performance of our system.

We show in our evaluations that the title of the source FAQ page is important in the retrieval efficiency of the system. Thus, the title supplies additional information for a question existing in the data collection. When the search results obtained by searching on *srcTitle* field compared with the search results obtained by searching on *question* field, searching on *srcTitle* field improved the retrieval performance of our system by 50%.

Also evaluations show that language should take in consideration in retrieval and the language specific analyzes should be done to improve the performance. This is the reason that Turkish Analyzer is more successful than Standard Analyzer.

For our data collection the evaluations show that, first retrieving the most relevant sources and then searching among the question answer pairs of these relevant sources for a given user question returns more relevant results than searching among all question answer pairs in the data collection.

Looking forward, we will expand our data collection and see whether first retrieving the source and then the question answer pair works well on large corpus. Also, to improve the retrieval performance of our FAQ retrieval system techniques such as question classes and part-of-speech tagging are going to be integrated into our system.

REFERENCES

[1] Wu C.H., Yeh J.F., and Lai Y.S., *Semantic Segment Extraction and Matching for Internet FAQ Retrieval*, IEEE Transactions on Knowledge and Data Engineering 2006, **18**(7), 930-940.

[2] Liljenback M.E., *CONTEXTQA: Experiments in Interactive Restricted-Domain Question Answering*, San Diego State University, 2007.

[3] Burger J., Cardie C., Chaudhri V., Gaizauskas R., Harabagiu S., Israel D., et al., *Issues, Tasks and Program Structures to Roadmap Research in Question Answering*, SRI International, 2003.

[4] Hirschman L., and Gaizauskas R., *Natural Language Question Answering: The View from Here*, Natural Language Engineering 2001, **7**(4), 275-300.

[5] Manning C.D., Raghavan P., and Schütze H., *An Introduction to Information Retrieval*, Cambridge, England: Cambridge University Press, 2007.

[6] Tomuro N., and Lytinen S.L., *Retrieval Models and Q and A Learning with FAQ Files*, New Directions in Question Answering, 183-202, 2004.

[7] Burke R., Hammond K., Kulyukin V., Lytinen S., Tomuro N., and Schoenberg S., *Natural Language Processing in the FAQ Finder System: Results and Prospects*, AAAI Spring Symposium on NLP on the WWW, Menlo Park, CA: AAAI Press, 17-26, 1997.

[8] Burke R.D., Hammond K.J., Kulyukin V.A., Lytinen S.L., Tomuro N., Schoenberg S., *Question Answering from Frequently Asked Question Files: Experiences with the FAQ Finder System*, University of Chicago, 1997.

[9] Camacho D., and Moreno M.D.R., *DynJAQ: An Adaptive and Flexible Dynamic FAQ System,* International Journal of Intelligent Systems 2007, **22**(3), 303-418.

[10] Hammond K.J., Burke R.D., Martin C., and Lytinen S.L., *FAQ Finder: A Case-Based Approach to Knowledge Navigation*. Proceedings of the 11th Conference on Artificial Intelligence for Applications: IEEE Computer Society, 1995.

[11] Sneiders E. *Automated FAQ Answering: Continued Experience with Shallow Language*, Understanding Procedings of the 1999 AAAI Fall Symposium, 1999.

[12] Radev D.R., Libner K., and Fan W., *Getting Answers to Natural Language Questions on the Web*, Journal of the American Society for Information Science and Technology 2002, **53**(5), 359-64.

[13] Whitehead S.D., *Auto-FAQ: An Experiment in Cyberspace Leveraging*, Computer Networks & ISDN Systems 1995, **28**(1-2), 137-146.

[14] Liu X., and Croft W.B., *Cluster-Based Retrieval Using Language Models*, Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, United Kingdom: ACM, 2004.

[15] Sheng-Yuan Y., *Developing an Ontological FAQ System with FAQ Processing and Ranking Techniques for Ubiquitous Services. First*, IEEE International Conference on Ubi-Media Computing, 31 Aug. 2008, 541-546.

[16] Jijkoun V., and Rijke M.D., *Retrieving Answers from Frequently Asked Questions Pages on the Web*, Proceedings of the 14th ACM International Conference on Information and Knowledge Management, Bremen, Germany: ACM, 2005.

[17] Hatcher E., and Gospodnetic O., *Lucene in Action*, Manning Publications, 2005.

[18] Arslan A., and Yilmazel O., *A Comparison of Relational Databases And Information Retrieval Libraries On Turkish Text Retrieval.* IEEE NLP-KE'08, Beijing, China, 2008.