Answering “Who is” Questions using Semantic Tag Clouds

GRADUATE PROJECT REPORT

Submitted to the Faculty of
The School of Engineering & Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX

in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Kiran Kumar Velineni
Summer 2013

Committee Members

Dr. Longzhuang Li
Committee Chairperson

Dr. Ajay Katangur
Committee Member

Dr. Scott A. King
Committee Member
ABSTRACT

Question Answering System (QAS) is a specialized area in the field of information retrieval, whose main objective is to obtain precise answers to a user’s questions. Most of the QAS deal with morphological and syntactic levels of processing. They concentrate on words or phrases or syntactic substructures of the sentences. Problems arise when there is a question such as “who is the outstanding player in 2010?” and an answer sentence of “Nadal is the prominent player in 2010” in the document set. A question answering system cannot retrieve the answer with only morphological and syntactic processing. Semantic level processing is required to match “outstanding” with “prominent”.

We present a Semantic QAS which deals with answering “Who is” questions using semantic tag clouds. The major goal of this project is to identify the answer sentences which contain semantically related question keywords and show the important role of semantic processing. Input of this project is a “who is” question and a group of documents. The system will process the documents, question and finds the main and possible answers to the question. The output is the answer for the “who is” question. The evaluated system produces answers to a set of “who is” questions, even when the question and answer sentences have different but semantically related keywords.
# TABLE OF CONTENTS

Abstract .............................................................................................................................. ii

Table of Contents ............................................................................................................... iii

List of Figures ...................................................................................................................... v

List of Tables .................................................................................................................... vii

1. Background ................................................................................................................1

   1.1 Introduction ........................................................................................................1

   1.2 Question Answering System .............................................................................3

   1.3 Morphological Analysis .....................................................................................4

      1.3.1 Baseline QAS ...........................................................................................5

   1.4 Syntactic Analysis ..............................................................................................6

   1.5 Semantic Analysis ..............................................................................................8

2. Problem Description .................................................................................................9

   2.1 Problem Statement .............................................................................................9

   2.2 Motivation ..........................................................................................................9

   2.3 Project Objective .............................................................................................10

   2.4 Project scope ....................................................................................................10

3. Semantic QAS Design ............................................................................................11

   3.1 How the Semantic QAS Works .......................................................................13

   3.2 Environment .....................................................................................................22

      3.2.1 NetBeans IDE ........................................................................................22

      3.2.2 Java Swing .............................................................................................22

      3.2.3 XAMMP ................................................................................................23
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1</td>
<td>Example Application of Document Retrieval</td>
<td>2</td>
</tr>
<tr>
<td>Figure 1.2</td>
<td>Question Answering System Architecture</td>
<td>3</td>
</tr>
<tr>
<td>Figure 1.3</td>
<td>A Parse Tree</td>
<td>7</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Flow Diagram of Semantic QAS</td>
<td>12</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Cricket.txt document about a Cricket Match</td>
<td>13</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>Example Paragraph Table</td>
<td>16</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Example Text File with 3 Paragraphs</td>
<td>24</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Sentence Splitting of First Paragraph as shown in Figure 4.1</td>
<td>25</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>An Example Processed Sentence</td>
<td>28</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>Paragraph table</td>
<td>30</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>Question Processed Sentence</td>
<td>32</td>
</tr>
<tr>
<td>Figure 4.6</td>
<td>Senses for “outstanding” in WordNet</td>
<td>33</td>
</tr>
<tr>
<td>Figure 4.7</td>
<td>Semantic Expansion of Keywords</td>
<td>33</td>
</tr>
<tr>
<td>Figure 4.8</td>
<td>Question Semantic Tag Cloud</td>
<td>34</td>
</tr>
<tr>
<td>Figure 4.9</td>
<td>Related Paragraphs for Sample Question</td>
<td>36</td>
</tr>
<tr>
<td>Figure 4.10</td>
<td>Related Sentences for Sample Question</td>
<td>37</td>
</tr>
<tr>
<td>Figure 4.11</td>
<td>Answers for Sample Question</td>
<td>38</td>
</tr>
<tr>
<td>Figure 4.12</td>
<td>User Interface with no indexed documents</td>
<td>39</td>
</tr>
<tr>
<td>Figure 4.13</td>
<td>User Selecting a Folder</td>
<td>40</td>
</tr>
<tr>
<td>Figure 4.14</td>
<td>Indexing User Selected Folder</td>
<td>41</td>
</tr>
<tr>
<td>Figure 4.15</td>
<td>Sample Question Posed by User</td>
<td>42</td>
</tr>
<tr>
<td>Figure 4.16</td>
<td>Answers obtained for the sample question with semantic expansion</td>
<td>43</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 1.1  Question and Answers returned by INQUERY .................................................5
Table 4.1  Treebank Tagset...............................................................................................27
Table 5.1  Evaluation Report............................................................................................47
1. BACKGROUND

This chapter discusses the information retrieval field, various applications of it, necessity of Question Answering System (QAS) and concepts of Natural Language Processing (NLP).

1.1 Introduction

Finding answers to user questions from giant databases, document sets, text and the World Wide Web is a very challenging task. Information retrieval is a research field which concentrates on finding user required information from different information sources. The applications of information retrieval are diverse, along with relevant information retrieval from large documents they also include, searching in digital libraries, automatic summarization, document classification and web searching [1]. Search engines are the most visible information retrieval applications. The traditional search engines have efficient and complex algorithms in providing answers to the user questions and they depend on keywords or patterns extracted from the query using natural language processing and data mining techniques [2].

In a traditional information retrieval system, the user first submits a query which is processed by the retrieval system. The latter, consults a database that contains document collection and returns the relevant documents. In order to process a user query or document, they should be transformed into a model understandable by an information retrieval system. Traditional information retrieval systems use a cosine vector space model to process a user query and documents. Every query and document is represented by a vector whose dimensions represent words that occur within them [3]. The value of
each dimension is the frequency of occurrence of that word in the document. The relevancy between the document and the query is measured by calculating the cosine of the angles between the query and document vectors. Based on the cosine similarity measures, relevant documents are retrieved from database. For example, Figure 1.1 shows a traditional search engine which retrieves a set of relevant documents in response to the question “who is the best tennis player in 2010?”

Figure 1.1 Example Application of Document Retrieval.
The relevant documents retrieved by a traditional search engine are large in number and a user has to manually browse the documents to find an accurate answer. So an information retrieval system that processes the relevant documents and gives precise answers to user questions is needed.

1.2 Question Answering System

Question answering system deals, with two different but related fields: information retrieval and natural language processing. On one hand, Information retrieval deals with large quantities of information, and on the other hand natural language processing is concerned with the understanding of a small piece of text. The general architecture of a question answering system consists of three modules: a question processing module, an information retrieval module and an answer extraction module [4]. Each module in turn has sub components as shown below in Figure 1.2.

![Diagram](figure1.2.png)

Figure 1.2 A Question Answering System Architecture.
In the question processing module, important keywords from the question are extracted and an answer type is determined. In the information retrieval module, relevant documents that contain the keywords extracted by the question processing module are retrieved. In the answer extraction module, the output from the information retrieval module and information about the expected answer type are used to determine the final answer. The final answer extraction step is where most question answering systems differ [5]. Question answering is the most challenging task in natural language processing. Unlike other natural language processing fields, which work on specialized tasks, question answering systems need to deal with the whole text and find which part of it contains the answer. In order to do this, a perfect question answering system should incorporate most other subfields of natural language processing, such as parts of speech tagging, parsing, named entity recognition, etc. The major task of a QAS is to achieve the human concept of “text understanding”. Natural language processing is divided into different levels, morphological analysis, syntactical analysis and semantic analysis.

1.3 Morphological Analysis

Morphological analysis is segmenting a sentence into a row of morphemes [6]. Morphology is the study of structure and formation of the word. During morphological analysis, words are considered separately to identify the simplest morphological components of them. One of the widespread tasks here is lemmatizing or stemming which is used in many web search engines. In this case, all morphological variations of a given word are collapsed to one lemma or stem. For example, a stemming algorithm reduces the words “fishing”, “fish”, “fished”, and “fisher” to the stem fish [6].
1.3.1 Baseline QAS

The Baseline question answering system contains three modules: question processing module, INQUERY search engine and answer extraction module. In the question processing module, classification and expected answer type of each question is determined. In the second module, the INQUERY search engine gets the query from the previous module, searches in its data collection and retrieves the top N documents. In the answer extraction module, answer candidates are extracted and their related scores are calculated using a scoring algorithm [7]. Four heuristics are considered in the scoring algorithm: number of matching query words, whether matching words are in the same sentence, best matching window size and the distance between the answer candidate and the center of the best matching window. Answer candidates are ranked based on their scores. But a problem with this question answering system is it only deals with the morphological level of processing. Let us consider an example shown in table 1.1 where three passages are returned by INQUERY to the question “who is the richest person in the world?”

<table>
<thead>
<tr>
<th>Question</th>
<th>Who is the Richest person in the world?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passage 1</td>
<td>Although tops in the U.S., Mr. Walton is the sixth-richest person in the world.</td>
</tr>
<tr>
<td>Passage 2</td>
<td>Once the richest black person in the world, Baker was destitute shortly before her death. She died in her sleep on the second night of a phenomenally successful comeback show in Paris.</td>
</tr>
<tr>
<td>Passage 3</td>
<td>As well as being the richest person in the world, Sir Hassanal lives with his relatives in the world’s biggest palace, a complex of buildings built with 38 types of marble on a 300-acre hill near the Brunei River. In case friends decide to stay over, it has 1,778 rooms and 257 toilets.</td>
</tr>
</tbody>
</table>
If “the richest person” is treated as bag of words, then, based on the heuristic rules Passage 1 and Passage 2 are considered as good passages and “Walton” or “Baker” is suggested as the answer. But actually, Sir Hassanal is the correct answer.

**Disadvantage:**

The Heuristics used in the baseline question answering system doesn’t make use of any syntactic information such as, how a question is phrased and the relationship between the words in the question. Neglecting crucial relations between words is a major source of false positives for current word based techniques [8].

### 1.4 Syntactic Analysis

Syntactic analysis extracts information such as parts of speech tags of words, phrases and the relationship between the words in the question and answer sentences. Tagging and parsing are generally used to extract syntactic information [9]. Parsing analyzes a sentence or string of words into its constituents, resulting in a parse tree. A parse tree represents the syntactic structure of a sentence according to some formal grammar. The parse trees are generally constructed in terms of constituency relations of the constituency grammar or in terms of dependency relations of the dependency grammar. The dependency based parse trees are simpler and contains fewer nodes than the constituency based parse trees. The dependency based parse tree lacks phrasal categories. Figure 1.2 shows a parser output, which is usually a tree like structure with a sentence label as the root, various phrase labels as intermediate nodes and words in the sentence as leaf nodes.
Many question answering systems have used the syntactic level of processing for obtaining correct answers to user questions and address the problem faced by morphological level of processing by matching dependency relations between the questions and answers. Katz et al. [8] used strict matching for matching dependency relations between question and answers. Cui et al. [10] performed fuzzy relation matching for matching dependency relation between question and answers. But the problem with these approaches [8, 10] are, dependency between the parts of a question and answer pair sentences are evaluated, without considering whether a passage contains an answer candidate or not. Sometimes, syntactic substructures frequently co-occur in sentences irrespective of whether a question and answer pair being true or not. Elif Aktolga et al. [11] have proposed an improved method that performs the candidate answer check by determining the answer type of the question. Passages are first passed through the initial check, and then the parse structure of question and answer pair can be analyzed with respect to the named entities.
Disadvantage:

Syntactic analysis provides answers to questions based on syntactic structure of a sentence, the relationship between the words and phrases match in both the question and answers. Syntactic analysis fails when a user asks a question in an indirect way or using different words from the words in the answer passage. For example if the user asks the question “who is the richest person?” and an answer passage has a sentence “Kevin is the wealthiest person in the world”, using syntactical analysis we cannot retrieve an answer to the above question. This level of processing contains two other sub-problems; Named Entity Recognition and Chunking or shallow parsing.

1.5 Semantic Analysis

The target of this level is processing the syntactic structures from the levels of phrases, clauses, sentences and paragraphs to their language independent meanings. Semantic analysis determines the possible meanings of a sentence by focusing on the interactions among word-level meanings in the sentence. This level covers the most complex task including: finding synonyms and word sense disambiguation. Basically, one has to complete morphological and syntactical analysis, before trying to solve any semantic problem.
2. PROBLEM DESCRIPTION

This chapter gives a detailed description of the problem statement, motivation, and project objective and project scope.

2.1 Problem Statement

Most of the existing question answering system deals with the problem depending on morphological and syntactic level of processing and retrieve precise text segments form the relevant documents. The components of natural language that they use, i.e., stemming, pos tagging, parsing does not always provide relevant answers. The problem is when there is a question like “who is the best player in 2010?” and an answer sentence of “Nadal is the most important player in 2010” is in the document set. Question answering systems dealing with only morphological and syntactic level of processing cannot retrieve the answer to the question.

2.2 Motivation

Millions of users search over the web to find answers to their questions related to different fields. The data is increasing exponentially and is becoming hard to convert the data into information. The traditional search engine retrieves a large number of relevant documents and user has to search all those documents to find the exact answer. A system that can process these documents that contain natural language text and provide an exact answer to a user question is important. Question answering is a computer science discipline that provides a correct answer to a user question, by processing the document text very similar to the human concept of “text understanding”.
2.3 **Project Objective**

The main objective of the proposed system is to provide answers to user “who is” questions, even when words in the question and the answer candidate sentences are different but semantically similar. This is done by semantically expanding the keywords in question and answer sentences by using WordNet [13]. In order to obtain the keywords various sub fields of natural language processing are applied on question and document sentences.

2.4 **Project Scope**

The input of this project is a “who is” question and a group of articles. The output of this project is the possible answers to a “who is” question. The proposed system deals with possible morphological and semantic differences between the question and answer. The proposed system preprocesses the given set of articles as input and indexes them in the database. The proposed system processes a user question, identifies keywords and expands them semantically. Finally, the proposed system will extract possible answers for user questions.
3. SEMANTIC QAS DESIGN

The semantic QAS consists of three main modules: indexing module (document preprocessing), question processing module and answer extraction module. In the indexing module, a set of documents is taken as input and the results are stored in the database. All the documents are preprocessed using Stanford CoreNLP [12], which is an integrated suite of NLP tools for English such as tokenization, lemmatization, POS tagging etc. Keywords and entities are extracted based on the preprocessing information and simultaneously are stored in the database. In this system, keywords are stored at document level.

In the question processing module, first, the question target type is determined. Second, the question sentence is processed using Stanford CoreNLP. Third, keywords and entities are recognized based on the preprocessing information. Fourth, extracted keywords are expanded semantically using WordNet [13]. WordNet is an open source free lexical database for English. Finally, a question semantic tag cloud that contains the question target type, semantically expanded keywords and entities of the question is saved.

In the answer extraction module, this system first does document filtering, which removes the irrelevant documents based on keyword matching and document weight. Document weight is calculated based on the question keyword frequency in documents. Second, it filters the paragraphs based on the paragraph score. Paragraph score is determined based on its keywords and entities matching with question keywords and entities. Third, the system also filters irrelevant sentences and gives scores to the relevant sentences. Finally, the system extracts the answers from the four top scored relevant
sentences from the sentence filtering step. The question target type determined in the question processing module is used to find the answer type and extracts the answer. Finally, possible answers are given as output based on their score. The flow diagram of the semantic QAS is shown in the Figure 3.1.
3.1 How the Semantic QAS Works

The semantic question and answering system has three modules a) indexing module b) question processing module and c) answer extraction module. If there are N documents and a “who is” question is posed to the system, it will first process all the documents one after another and store them into the database. Secondly, it analyzes the question in question processing module and extracts answer in the answer extraction module. For testing, question “who is the outstanding player in the match?” is posed to the system. Cricket.txt document has the information about India versus Pakistan match as shown in Figure 3.2.

Figure 3.2 Cricket.txt document about a Cricket Match.
Indexing module:

First system preprocesses the cricket.txt file and stores the results in the database using the following steps:

Step 1: System extracts the text out of the file and splits it into paragraphs based on the new line character. In our cricket.txt document we have 3 paragraphs. Each paragraph is processed one by one.

Step 2: For example, consider the third paragraph in Figure 3.2. SentenceAnnotations class from Stanford CoreNLP is used to split the paragraph text into sentences. By the end of this step, we get four sentences for the example paragraph.

Step 3: Each sentence is processed using the Stanford CoreNLP components such as tokenization, lemmatization and POS tagging. Tokenization splits a sentence into tokens, nothing but words. For example, consider the fourth sentence form third paragraph in Figure 3.2 “For scoring a match winning hundred, Dhoni is awarded as a prominent participant”. After tokenization, the example sentence is broken into 15 tokens.

Step 4: Now, each token is processed by lemmatization and POS tagging. Lemmatization reduces a given word to its lemma. POS tagging assigns the parts of speech to the token. Detailed explanation of Lemmatization, POS tagging is provided in section 4.1.3. So, once all the tokens are processed, then the processed sentence contains the following information: tokens, each token lemma and POS tag. The following is the output for the sentence: [lemma/POS tag]

Step 5: In this step stop words will be removed and only keywords and entities are considered using the code shown in Appendix Figure A.1. Basically words whose POS tag starts with the following letters: N, CD, V, J, and R are considered as keywords. Words with POS tags NNP and CD are taken as entities. NNP is a singular Noun and CD is cardinal number. More information about POS tags is shown in table 4.1. For the example sentence considered, after completing this step following is the output

Keywords: participant, prominent, win, award, Dhoni, match, two, hundred

Entities: two, hundred, Dhoni

Step 6: The extracted keywords are inserted into the database table’s keywords and keyword_document. If the keyword is already in the keyword_document table, its frequency of occurrence is increased.

Step 7: Each sentence of the paragraph is processed similarly using steps 3,4,5,6. Once all the sentences of the paragraph is processed. The paragraph table in the database is updated. The following Figure 3.3 shows the paragraph table of example
paragraph in Figure 3.2. The paragraph table contains paragraph id, document id, paragraph content, keywords and entities of the paragraph.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Function</th>
<th>Null</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>parg_id</td>
<td>int(11)</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>doc_id</td>
<td>int(11)</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>parg_content</td>
<td>text</td>
<td></td>
<td></td>
<td>After the completion of the match. In post match presentation, the two team captains and the teams participated. Winning captian Dhoni lift the cup for india. For scoring a match winning hundred, Dhoni is awarded as a prominent participant.</td>
</tr>
<tr>
<td>keywords</td>
<td>text</td>
<td></td>
<td></td>
<td>#cup:#post:1#captian:1#Dhoni:1#hundred:1#score:1#bonus:1#award:1#team:2#win:2#completion:1#captain:1#two:1#indium:1#presentation:1#lift:1#participant:1#prominent:1#match:3#participate:1#</td>
</tr>
<tr>
<td>entities</td>
<td>text</td>
<td></td>
<td></td>
<td>#two#hundred#Dhoni#</td>
</tr>
</tbody>
</table>

Figure 3.3 Example Paragraph Table.

Step 8: Similarly all the paragraphs in the example Cricket.txt document are processed using the steps 2, 3, 4, 5, 6, 7. Once the file is processed, the entire database table’s keyword, documents, keyword_document and paragraph are updated. The keywords table has id and keywords as fields. The documents table has id, doc_name, doc_path as fields. The paragraph table has parg id, doc_id, parg content, keywords and entities as fields and keyword_document table has id, doc_id, keyword_id, freq and weight.
**Question Processing module:**

In this module the example question “who is the outstanding player in the match?” is analyzed by the following steps:

*Step 1:* In this step question target type is determined. In this project we are concentrating on two types of “who is” questions. First type is, our example, where target type is a person. Second type, if there is a question like “who is Kevin?” the target type of it will be description about the person. By the end of this step we will determine the question target type based on the code shown in appendix Figure A.4. For our example question target type is a person.

*Step 2:* After obtaining the target type, the question is processed using the indexing module steps 3, 4, 5. Keywords and entities are obtained by the end of this step. Following are the keywords and entities for the example question:

*Keywords: outstanding, player, match*

*Entities: no entities, since no singular noun or cardinal number in the question.*

*Step 3:* In this step question keywords are semantically expanded using WordNet. It is a lexical database for English. It provides synonyms for every keyword. By the end of this step, example question has the following expanded semantic keywords.

*Outstanding: prominent, salient, spectacular, striking, owed owing, undischarged*
Player: player, participant, instrumentalist, musician, historian, actor

Match: equal, mate, couple, playoff, peer, compeer

By the end of this module, each question has its question target type, semantically expanded keywords and entities. For the example question, we have target type as person and semantically expanded keywords.

Answer Extraction Module

In this module, main and possible answers are extracted using the following steps:

Step 1: If there are n documents, then relevant documents to the user question are determined based on the matching keywords weight. Each question keyword is compared with the document level keyword set. If it finds a match then the document weight is increased by matching the keyword weight in the document. Finally, each document has a weight, based on which documents are sorted. If no question keywords are matched with document level keywords, then that document is eliminated. By the end of this example document cricket.txt has weight of 7. Because, the weights of the matched keywords are outstanding, player, match is 5, 1, and 1 respectively.

Step 2: In this step the question keywords and entities are matched with paragraph level keyword and entities of the document. For example, we have three paragraphs. First, if a question contains entities, then each paragraph level entities are compared with the question entities, if all the entities are matched then the paragraph is sent to next step. If
there are no matches then the paragraph is removed. Since there are no entities for the
example question, it skips to paragraph level keyword check. Question keywords are
checked with the paragraph level keywords. If there is any match, then the paragraph
score is increased by matching keyword frequency. Once all the question keywords are
checked, each paragraph will have its score. The paragraph keywords should have at least
n-1 matching question keywords, if not, the paragraph will be removed. In our example
document we have 3 paragraphs. By the end of the step, each paragraph has score 2, 1
and 5 respectively. The first two paragraphs will be removed because out of 3 question
keywords they just have 1 matching keyword in their paragraph. So, only third paragraph
in Cricket.txt document is sent to next step.

Step 3: In this step, related sentences of the relevant paragraphs are determined. Each
sentence is checked for question entities, if it contains the question entities, then the
sentence is sent to keyword level matching. If not, the sentence is removed. Each
sentence is checked for question keyword. If it finds a match then its score gets increased
by 1. Once all the sentences are checked. First four sentences with highest scores are sent
to the next phase. In this example, by the end of this step, paragraph three contains four
sentences with its score. Since sentence 3 has no score it is eliminated. Remaining three
sentences 1, 2, 4 will be sorted according to their score and sent to final phase.

1) After the completion of the match. (Score 1)

2) In post match presentation, the two team captains and the teams participated.
   (Score 1)

3) Winning captain Dhoni lift the cup for india. (Score 0)
4) For scoring a match winning hundred, Dhoni is awarded as a prominent participant. (Score 3)

Step 4: Four highest scored sentences are obtained from the previous step. If all the obtained sentences have different score, then main answer is extracted from the sentence with highest score and other answers are extracted from remaining sentences. If the sentences have same score, and then based on the paragraph score the main answer and other possible answers are extracted. In order to extract the answer from sentence, first we will check our question target type and then extract the answer. In our example, the following are the sorted sentences obtained from previous phase.

1) For scoring a match winning hundred, Dhoni is awarded as a prominent participant. (Score 3)
2) After the completion of the match. (Score 1)
3) In post match presentation, the two team captains and the teams participated. (Score 1)

Sentence 1 has the highest score (3). So, main answer is extracted from this sentence and other answers from the remaining sentences. Since example question target type is person, we first get the words that start with capital letter from each sentence. Later, it checks whether the extracted word is a person or not, is done by checking in the database table person_name_gazetter. We depend on persons dataset presented by
Conference on Computational Natural Learning (C0NLL - 2003) to confirm the extracted information is a person [24].

1) Dhoni is awarded as a prominent participant. (Dhoni)

2) After the completion of the match. (After)

3) In post match presentation, the two team captains and the teams participated. (In)

Dhoni, After, In are extracted from the three sentences. Now they are checked in the database table person_name_gazetter to confirm whether it is a person or not. So, in this example, sentence 1 has Dhoni as person and it has highest score. So, it is provided as main answer. Its percentage is calculated by (answer score * 100) divided by sum of answer scores. Here answer score is nothing but sentence score. Since in this example we have Dhoni as the only answer, the percentage is (3 * 100) / 3. So Dhoni is provided as main answer with percentage of 100. The major aim of this project is to identify the answer to a “who is” question, even when the answer and question sentences has different but semantically related keywords. Second aim is to utilize the natural language processing techniques to the maximum. In our example, even the main answer sentence has different keywords (prominent, participant) which are semantically related to the question keywords (outstanding, player). Semantic QAS system is successfully identifying them.
3.2 Environment

The system was developed using the java programming language. Java swing, netbeans and mysql are mainly used for the implementation of this project. In order to run the applications developed by the NetBeans IDE, the Java Runtime Environment has to be setup in the computer. Java SE Development Kit 7 [23] contains the complete JRE and tools, is used in this project.

3.2.1 NetBeans IDE

NetBeans is an Integrated Development Environment (IDE) used to develop mobile, desktop and web applications in Java [14]. It can be installed on any operating system that supports java. NetBeans IDE 7.2 version is used in this project. NetBeans provides various features such as efficient project management, fast user interface development, debugger, source code editor. It has well documented APIs and is extensible. NetBeans debugger allows you to add field watches, break points in source code, run into method and monitors execution. It is an open source cross-platform IDE.

3.2.2 Java Swing:

Java swing is a Graphical User Interface (GUI) toolkit which enables java programmers to create GUI for their desktop, web or mobile applications [15]. It is a part of the java foundation classes. Java Swing provides more advanced GUI components than Abstract Window Toolkit (AWT). Java Swing components are written in java and they are platform independent. Along with the familiar components such as buttons and labels, java swing also provides several sophisticated components such as, scroll panes, lists, trees and tables. Javax.Swing package has all swing components and classes. All the class names begin with ‘J’ such as JButton and JList. Few top level swing classes such as
JFrame, JDialog and JWindow are said to be heavily weight as they extend the AWT versions.

3.2.3 XAMPP

XAMPP stands for “X, Apache, Mysql, PHP, Perl” and is a bunch of software that installs each of this items [16]. XAMPP 1.5.5 version is used in this project. XAMPP also contains other modules such as phpMyAdmin and OpenSSL. XAMPP enables a user to create and manipulate databases in mysql using phpMyAdmin. phpMyAdmin is an open source Mysql administrating tool written in PHP. It is signified to manage the administration of mysql with the help of web browser. phpMyAdmin enables a user to perform tasks such as, creating, modifying and deleting tables, rows, fields, manage user permissions and execute SQL statements.

3.2.4 Stanford CoreNLP

Stanford CoreNLP is an integrated framework that provides set of natural language analysis tools which takes raw English text input and gives the base form of words, their parts of speech, markup the structure of sentence in terms of phrases and word dependencies [12]. The Stanford CoreNLP is written in java and it requires java version 1.6 or greater. The major goal of this framework is to enable people to quickly get complete linguistics annotations of natural language text. This is highly flexible and extensible. It is an open source code; we can even add our own annotators.
4. IMPLEMENTATION

This chapter gives a detailed description of how semantic QAS modules such as indexing, question processing and answer extraction modules are implemented.

4.1 Indexing Module:

In this module, document repositories are processed one after another and updated to the database. The key purpose of storing an index is to optimize speed and performance in finding relevant answers for a user requested question. Let’s consider a scenario, where the text file Figure 4.1 has 3 paragraphs. The example file is preprocessed and the results are stored in the database as shown in the following sub sections. The database consists of four different tables named documents, keywords, keyword_documents, and paragraphs. Once a file has been preprocessed, all the tables get updated with relevant information. For example, processing and indexing of a text file in Figure 4.1 is shown in the following steps.

Figure 4.1 Example Text File with 3 Paragraphs.
4.1.1 File Processing

The text extraction from a file is processed starting with paragraph identification. Based on newline character, the document is processed by separating paragraphs. From Figure 4.1, we now have three paragraphs. Now, each paragraph of the text document is considered one after another for processing and indexing.

4.1.2 Paragraph Processing

During this phase, the output from file processing is considered as input. CoreAnnotations.SentenceAnnotations class is used to split the paragraph text into sentences. This Annotation class is used from the edu.standford.nlp.ling package [17]. The Figure 4.2 shows the sentences of first paragraph in Figure 4.1. After splitting, each sentence of the paragraph is sent to sentence processing one after the other.

![Figure 4.2 Sentence Splitting of First Paragraph as shown in Figure 4.1.](image)

4.1.3 Sentence Processing

During this phase, the output from paragraph processing is utilized as input. The sentences obtained from the earlier phase are processed using the Stanford CoreNLP components such as tokenization, lemmatization and parts of speech tagging.

Tokenization: Tokenization is the process of breaking a sentence into words, phrases, meaningful elements or symbols called tokens. In this step, sentence is tokenized into words using CoreAnnotations.TokenAnnotations [18]. The above annotation class is used from edu.standford.nlp.ling package. For example consider the second sentence of the
first paragraph of Figure 4.1 i.e., “Raju is the prominent tennis player in 2010”. After applying the tokenization, 9 tokens are obtained. Now, lemmatization and parts of speech tagging is applied on each and every token of the input sentence.

*Lemmatization:* Lemmatization is the process of reducing a given word to its lemma [19]. It aims at removing inflectional endings and return dictionary or base form of word called lemma. Lemmatization is closely related to stemming. The difference is, stemming operates on a single word without the knowledge of context whereas lemmatization operates on a single word with the use of vocabulary and morphological analysis of words. If stemming and lemmatization are applied on the word “saw”, stemming returns letter s, whereas lemmatization returns see/saw. In this step, each token from the previous step is taken and its lemma is determined. This step is performed using CoreAnnotations.TokenAnnotations [20] class which is a part of standfordnlp.ling package. Lemmatization reduces the number of inflected and derivative words used to represent documents by mapping to same stem, which saves storage space and time taken to process them.

*POS Tagging:* Parts of speech tagging is the process of assigning parts of speech to each token such as noun, adjective, verb, adverb etc. Each token from earlier phase is taken and their parts of speech are determined. CoreAnnotation.PartsOfSpeechAnnotation class which is a part of the standfordnlp.ling package is used to perform this task [21]. Parts of Speech annotator of corestandfordnlp software uses tags from Penn Treebank tagset [22]. The following table shows a few important tag sets of Penn Treebank tagset.
Table 4.1 Treebank Tagset.

<table>
<thead>
<tr>
<th>POS Tag</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>coordinating conjunction</td>
<td>And</td>
</tr>
<tr>
<td>CD</td>
<td>cardinal number</td>
<td>1, third</td>
</tr>
<tr>
<td>DT</td>
<td>Determiner</td>
<td>The</td>
</tr>
<tr>
<td>EX</td>
<td>existential there</td>
<td>there is</td>
</tr>
<tr>
<td>FW</td>
<td>foreign word</td>
<td>d’hoever</td>
</tr>
<tr>
<td>IN</td>
<td>preposition/subordinating conjunction</td>
<td>in, of, like</td>
</tr>
<tr>
<td>JJ</td>
<td>adjective</td>
<td>Green</td>
</tr>
<tr>
<td>JJR</td>
<td>adjective, comparative</td>
<td>Greener</td>
</tr>
<tr>
<td>JJS</td>
<td>adjective, superlative</td>
<td>Greenest</td>
</tr>
<tr>
<td>LS</td>
<td>list marker</td>
<td>1)</td>
</tr>
<tr>
<td>MD</td>
<td>Modal</td>
<td>could, will</td>
</tr>
<tr>
<td>NN</td>
<td>noun, singular or mass</td>
<td>Table</td>
</tr>
<tr>
<td>NNS</td>
<td>noun plural</td>
<td>Tables</td>
</tr>
<tr>
<td>NNP</td>
<td>proper noun, singular</td>
<td>John</td>
</tr>
<tr>
<td>NNPS</td>
<td>proper noun, plural</td>
<td>Vikings</td>
</tr>
</tbody>
</table>

By the end of the sentence processing, each processed sentence contains processed tokens. Each processed token in turn contains lemma and POS tagging of it.

For example, the following sentence is given as input for the sentence processing step,
“Raju is the prominent tennis player in 2010”. Output of the processed sentence is shown in the following Figure 4.3.

![Figure 4.3 An Example Processed Sentence.](image)

### 4.1.4 Keywords and Entities Extraction

In this step, keywords and entities are extracted from the processed sentences of each paragraph. The major task of this step is to eliminate the stop words and get important words of the processed sentences which are keywords. Based on the morphological and syntactical information of the processed sentences, we get the keywords. The code in Figure A.1 eliminates the stop words and finds the keywords and entities in the processed sentences.
In Figure A.1 lines 54 and 55 is used to extract the keywords and eliminate stop words from each processed sentence. Processed tokens which are verb, adjective, and adverb are taken as keywords. The obtained keywords are inserted into the keywords table in database using the code in Figure A.2.

In order to make a relation between the keyword and document and insert it into the keyword_documents table, code in Figure A.3 is used. It first checks relation between the document and keyword, if it exists it just increases the frequency of the keyword else it will insert the new relation in keywords_documents table.

Line 65 and 66 in Figure A.1 shows the code based on which we extract the entities from the processed sentences. The processed tokens already have the information about their POS tags, so processed tokens whose POS tags are NNP and CD are extracted as entities. NNP is a singular noun and CD is a cardinal number, more information about POS tags is provided in table 4.1 Treebank Tagset.

After obtaining the keywords and entities, finally document id, paragraph content, keywords and entities are inserted in the paragraph table in the database.
Figure 4.4 shows Paragraph table contents. In this way, each paragraph of the document will be processed. Once all the files have been processed, the weight of the document level keywords are determined and saved by the fill weight method. All keywords have to be considered equally important when it comes to assessing relevancy on a query. So we need to reduce the frequency weight of a keyword by a factor that grows with its collection frequency [19]. If there are N documents in a document collection and a keyword Ki occurs M times. Then N divided by M is the inverse document frequency. The keyword weight will be keyword frequency multiplied by inverse document frequency.
4.2 Question Processing Module

In this module a user submitted question undergoes the following outlined subsections:

- 4.2.1 Question Target Type
- 4.2.2 Question Processing
- 4.2.3 Question Keyword and Entities Extraction
- 4.2.4 Semantic Expansion of Keywords

4.2.1 Question Target Type

In this step, the question target type is determined. There are many types of questions such as who, where, when etc. Who questions targets persons, where questions targets places, when questions targets date and time. This project aims at answering “who is” questions. Figure A.4 shows the code used to determine the question target type. By the end of this step the question target type is determined.

4.2.2 Question Processing

In this step, question sentence is processed through the same procedure provided in subsection 4.1.3 in indexing module. For example if we have a question such as “who is the outstanding player in 2010?” Figure 4.5 is the output of the processed question sentence. The example question sentence is broken into tokens and each token has its lemma and parts of speech. Based on this information the keywords and entities of the question are determined.
4.2.3 Question Keywords and Entities Extraction

In this step, keywords and entities are extracted from the processed question sentence by following the same procedure in step 4.1.4 in the indexing module. For the example taken in previous step, the keywords and entities obtained are “outstanding, player”, “2010” respectively.

4.2.4 Semantic Expansion of Keywords

In this step the keywords obtained from the previous step are semantically expanded using WordNet. It is a lexical database of English where nouns, adjectives, adverbs and verbs are grouped into Synsets [13]. WordNet 2.1 version is used in this project. It interlinks word forms along with specific sense of words. It finds the semantic relationship between the words. WordNet has 117000 Synsets. For example, the term outstanding has 4 senses in Figure 4.6.
Figure 4.6 Senses for “outstanding” in WordNet.

The keywords obtained in previous step are semantically expanded and their output is shown in Figure 4.7.
Finally after the four subsections, the question target type and semantically expanded keywords and entities of the given question sentence are saved in an object “sq”. This is shown in Figure 4.8.

![Figure 4.8 Question semantic tag cloud.](image)

### 4.3 Answer Extraction Module

In this module, the output from the question analysis phase (that contains question information) is taken as input. Based on the input, the answer extraction is done through the following steps:

#### 4.3.1 Document Filtering

In this phase, documents related to question are determined based on the question semantic tag cloud. Question keywords are sent to the database. Each question keyword is compared with the document level keyword set. If it finds a match then that document name and keyword weight will be returned. Keyword weight is obtained from the fill weight method in indexing module. If no question keywords match with document level keywords then that document is eliminated. By the end of this phase documents name
and weight are obtained. Based on their weights documents are sorted and sent to the next phase.

4.3.2 Paragraph Filtering

In this Phase, the output from the document filtering phase and question semantic tag cloud are taken as input. Based on the obtained input files, the corresponding paragraphs in the input files will be retrieved. Paragraphs contain the keywords and entities. In order to find the related paragraphs following two steps are done:

Paragraph Level Entity Matching: If a question semantic tag cloud contains entities, then each paragraph level entities are compared with the question entities, if all the entities are matched then the paragraph is sent to next step. If there are no matches then the paragraph is removed. For example, if there is a question such as “who is the outstanding player in 2010?” and a document containing three paragraphs as shown in Figure 4.1, by the end of this step, the paragraph starting with “Kevin” is removed, because it doesn’t contain the entity “2010”.

Paragraph Level Keyword Matching: In this step, Paragraphs obtained from last step are taken as input. Each question keyword set is compared with the paragraph level keywords. Whenever a matching keyword is identified, then the paragraph score is increased by matching keyword frequency at the paragraph level. By the end of this step each paragraph obtains a score based on their matching keywords.

After going through this phase, the example in Figure 4.1 gets the output shown in Figure 4.9. Each related paragraph contains doc id, paragraph id, paragraph keywords, paragraph entities and paragraph score.
4.3.3 Sentence Filtering

In this step, related paragraphs obtained in the previous step and question semantic tag cloud are taken as input. In order to find the related sentences from the related paragraphs, the following two steps are performed:

*Sentence Level Entity Matching*: If a question semantic tag cloud contains entities, then each sentence entities are compared with the question entities, if all the entities are matched then the sentence is sent to next step. If there are no matches then the sentence is removed.

*Sentence Level Keyword Matching*: In this step, each question keyword set is compared with the sentence level keywords. Whenever a matching keyword is identified, then the sentence score is increased by one. By the end of this step each sentence obtains a score based on their matching keywords.
Finally, by the end of step 2, each sentence obtains its score. Based on the scores, sentences are sorted and the four highest scored sentences are sent to the answer extraction phase. After going through this phase, the example in Figure 4.1 gets the output shown in Figure 4.10

![Figure 4.10 Related Sentences for Sample Question.](image)

### 4.3.4 Final Answer Extraction

In this phase, the four highest scored sentences from the previous phase are taken as input. Now, the answer type is determined based on the question target type obtained in the question processing module. For example, if the question starts with “who is” then its answer type is determined as person. Now based on the answer type the relative information is extracted from the answer sentence. We depend on persons dataset
presented by Conference on Computational Natural Learning (CONLL - 2003) to confirm the extracted information is a person or not [24]. This is other way of confirming the named entity. If the related sentence scores are different, then the answer extracted from the highest scored sentence is provided as the main answer and the remaining are provided as possible answers. The percentages of the main and possible answers are obtained by sum of related sentence scores divides related sentence score times 100.

If the scores of the related sentences are the same, then based on the paragraph scores of the related sentences, the main answer is extracted. The answer scores of each related sentence is multiplied with its paragraph score. Now, based on the obtained answer scores, the percentages will be determined for each answer and finally the main and possible answers are provided as output. The following Figure 4.11 shows the possible answers for the sample questions.

![Figure 4.11 Answers for Sample Question.](image)
4.4 User Interface Module

Java Swing, which is a part of java foundation classes, is used to build the user interface. The following figures shows how a user interacts with semantic QAS using the user interface and obtains answer to “who is” question. Figure 4.12 shows user interface of semantic QAS.

Figure 4.12 User Interface with no indexed documents.
Figure 4.13 shows the user selecting a folder that contains documents to be indexed. Before going to pose questions, a user has to index the folder that contains documents. Each document will be preprocessed by the indexing module and results are stored in the database.

Figure 4.13 User Selecting a Folder.
Once the user selects the folder and clicks open, the documents in the selected folder will be preprocessed one by one in the indexing module. A progress bar will indicate the status of the preprocessing documents. Figure 4.14 shows the progress bar of the documents being indexed. By the end of indexing all the documents get processed and stored in the database.

Figure 4.14 Indexing User Selected Folder.
Once the documents in the folder are indexed than the user can start posing “who is” questions. Figure 4.15 shows a sample question posed by the user. A user can pose the sample question in two different ways, with selecting and without selecting the “use semantic” check box. The question posed by the user will be analyzed by the question processing module and provides valuable information to answer extraction modules.

Figure 4.15 Sample Question Posed by User
Based on the information provided by the question processing module, the answer extraction module retrieves the appropriate answers for the user question. It undergoes the document filtering, paragraph filtering, sentence filtering and finally extracts main and possible answers from the top four highest scored relevant sentences. Figure 4.16 shows the main answer and other answers for the user sample question with semantic expansion.

Figure 4.16 Answers obtained for the sample question with semantic expansion.
If a user question is posed without selecting the “use semantic” check box, then the search for the answers is done without expanding the question keywords semantically. Figure 4.17 shows the main answer and other answers for the user sample question without semantic expansion.

Figure 4.17 Answers obtained for the sample question without semantic expansion.
If the semantic QAS doesn’t retrieve any answers than it will show no possible answers as shown in Figure 4.18 Indexed document count is represented at the bottom with name “index content”, which indicates the number of documents got preprocessed. In the considered example, since we have only one document, we can see the index content represented by one.

Figure 4.18 No Possible Answers.
If a question has been posed without preprocessing the documents, then an error message is indicated as “please index file before posting a question” as shown in Figure 4.19. A user has to preprocess the documents before posing a question. If documents are not indexed, then the index content is 0 as shown in the Figure 4.19.

Figure 4.19 Please Index Document before Posting a Question.
5. TESTING AND EVALUATION

This application is evaluated with around 100 different questions posed on 6 documents. Each document has different information such as about American presidents [29], tennis players [30], inventors [25], economics [27], quantum computers [26] and Nobel Prize [28]. A question answer dataset of 100 questions and their answers were determined manually on those documents. The first experiment was to run our Semantic QA system with these 100 questions without involving the semantic expansion. In the latter case, we did the same involving the semantic expansion. In the first experiment, system gave correct answers for 65 questions and gave wrong answers for 35 questions. In the second experiment, system gave correct answers for 77 questions and wrong answers for 23 questions. The evaluation information is represented in tabular form, shown in Table 5.1.

<table>
<thead>
<tr>
<th>System/Answers</th>
<th>Correct</th>
<th>Wrong</th>
</tr>
</thead>
<tbody>
<tr>
<td>With Semantic</td>
<td>77</td>
<td>23</td>
</tr>
<tr>
<td>Without Semantic</td>
<td>65</td>
<td>35</td>
</tr>
</tbody>
</table>

The evaluation report about how many questions were posed on each document and results of them are represented in an excel worksheet. This is shown in figure 5.1 which has three columns, first column contains question id, second column containing results of the QA system using semantic expansion and third column contains results of the QA system without semantic. A value ‘true’ represents that the answer pulled by QA system is correct and a wrong answer is represented by ‘false’.
Figure 5.1 System output for the questions posed

The objective of this evaluation is to study the results obtained by following a particular approach i.e., using semantic information in retrieving answers from the question and answering system.
**Evaluation with Huge Data:** The semantic QAS system has also been tested on a word document that contains 120 pages of data about 12 American presidents and 6 tennis players. American president’s information such as their introduction, early life and political carrier information is taken from Wikipedia [29]. Tennis player’s information such as their introduction, childhood and tennis carrier is taken from Wikipedia [30].

After indexing the large document, 20 questions (provided in Appendix B2) were posted on the system. Semantic QAS system gave correct answers for 14 questions and wrong answers for 6 questions. The following Example1, Example2, Example3 gives brief explanation of cases in which the system provided wrong answers and reasons for it.

Example1, Question such as “who is the sixth president of the United States?” got main answer as Abraham Lincoln, instead of John Quincy Adams. The reason is since the sentence score of both the persons is same, so based on the paragraph score main answer is provided (Abraham Lincoln paragraph has high score than John Quincy Adams).

Example2, Questions such as “who served as the 42nd president of the United States?” got no possible answers. The reason is even though the related sentence (contains answer William Jefferson Clinton) is obtained, the name obtained from sentence is not founded in the database table person_names_gazetter. Example3, consider a question such as “who is the president of United States?” And an answer sentence such as “Obama and Clinton are the president and vice president of United States”. Semantic QAS cannot answer this example3 questions, the reason is it cannot identify the relationship or dependencies between the words.
The objective of this evaluation is to study how the Semantic QAS is providing results when large data has been given as input. As the data increases the preprocessing time is increasing.

The developed Semantic QAS is tested on the following functionalities:

- Allowing a user to select folder from desktop
- Preprocess the documents and store the results in database
- If a question is posed without documents processed, semantic QAS system give error message “please index file before posting a question” as shown in Figure 4.19.
- Allowing a user to get answers with and without involving semantic expansion as shown in Figures 4.16 and 4.17.
- If a question doesn’t retrieve any answers the application will give a response “no possible answers” as shown in Figure 4.18
- Whether retrieving relevant documents, paragraphs and sentences of a given question or not
6. CONCLUSION AND FUTURE WORK

The semantic question answering system developed provides answers to "who is" questions. This was achieved in three different steps. In the first step, the system processes the documents provided by the user and stores the results in a database at document level, paragraph level, and sentence level. In the second step, the system processes the user question by semantically expanding the keywords obtained. In the last step, using the question information as input, the database is searched for related documents, paragraphs, and sentences. We have seen that in providing results, the system retrieves main answer and likely possible answers based on their scores. The resources mainly used to develop this system are Stanford CoreNLP for tokenization, lemmatization, and parts of speech tagging, WordNet for enhancing the question keywords semantically. This system is successful in identifying the answer sentences, even when the question and answer sentence keywords are different but semantically related.

Apart from the work done towards this system, the system can be further enhanced in future with below listed objectives in providing answers to the user questions:

- Extending to answer question types such as what, when, where etc.,
- Improve the retrieval time when dealing with huge amount of data.
- Increase the accuracy, using document processing techniques.
- Making it a real time application, by getting the related files to user question from third party such as Google.
BIBLIOGRAPHY AND REFERENCES


Appendix A: Code Snippets

A.1 Explanation of Code Snippets

This appendix A shows the screen shots of the code used in the project at different modules.

Figure A.1 shows how the keywords and entities are extracted during the text processing in indexing module.

```java
int file_id = DB_Operations.getDocumentId(file.getName(), file.getPath());
for (int i=0; i<as.size(); i++) {
    Processed_sentence ps = as.get(i);
    for (int j=0; j<ps.tokens.size(); j++) {
        Processed_token t = ps.tokens.get(j);
        // checking whether a word is keyword or not
        // based on the information obtained from Lemmatization and POS Tagging keywords extraction:
        // Words with Parts Of Speech Adjective, Noun, Adverb, Verb and Cardinal Number are taken as Keywords.
        if (t.lemma.equals("be") || t.pos.startsWith("NP") || t.pos.startsWith("CD") || t.pos.startsWith("V")) {
            AddKeywordToIndex(t.lemma, file_id);
            if (tags.containsKey(t.lemma)) {
                tags.put(t.lemma, tags.get(t.lemma)+1);
            } else {
                tags.put(t.lemma, 1);
            }
        }
        // Words which has POS "NNP" and "CD" are taken as Entities.
        // In most cases these words are nouns and numbers.
        if (t.pos.startsWith("NNP") || t.pos.startsWith("CD")) {
            entries += t.lemma + ";
        }
    }
}
```

Figure A.1 Keyword and Entity Extraction.
The code used for insertion of keywords into the keywords table is shown in following Figure A.2.

```java
//create a connection
dBConnect.getConnection();
//query to check the existence of the keyword
ResultSet resultSet = dBConnect.executeQuery("select * from keywords WHERE keyword='"+keyword+'"');
try {
    //get results
    while (resultSet.next()) {
        // if the keyword is exist get its id and return it.
        int keywordId = resultSet.getInt(1);
        return keywordId;
    }
    //if not exist insert the keyword
    int res = dBConnect.executeUpdate("insert into keywords (`keyword`) VALUES ('"+keyword+'")");
    if(res>0)// and then return its id
        return getWordId(keyword);
    else return -1;
} catch (SQLException ex) {
    Logger.getLogger(DB_Operations.class.getName()).log(Level.SEVERE, null, ex);
    return -1;
}
```

Figure A.2 Keywords insertion into database.
Figure A.3 shows the code used for updating the Keyword_document table in the database.

```java
// check whether the relation is existed or not
ResultSet resultSet = dBConnect.executeQuery("select * from keyword_documents WHERE doc_id="+doc_id+'";
try {
    while (resultSet.next()) {
        // if relation is existed, freqeuncy is incremented
        int id = resultSet.getInt(1);
        int res = dBConnect.executeUpdate("update keyword_documents set freq=freq+1 where id="+id+'");
        if(res>0)
            return;
    }
    // if relation does not existed, then insert it
    int res = dBConnect.executeUpdate("INSERT INTO `keyword_documents` (doc_id, `keyword_id`, `freq"
) catch (SQLException ex) {
    Logger.getLogger(DB Operations.class.getName()).log(Level.SEVERE, null, ex);

Figure A.3 Keyword_Document relation.

Figure A.4 shows the code used for determining the target type of the question.

```java
SearchQuery(String question) {
    this.question = question;
    if(question.toLowerCase().startsWith("who"))
        target_type = Question_Types.WHO_IS;
    if(question.toLowerCase().startsWith("where"))
        target_type = Question_Types.WHERE;
}
```

Figure A.4 Question Target Type.
Appendix B: Evaluation Questions

B.1 Evaluation Questions for 6 Documents

1. Who is James Madison?
2. Who was the fourth president of the United States?
3. Who was the quaternary president of the United States?
4. Who was the fifth president of the United States?
5. Who was the sixth president of the United States?
6. Who was the second president of the United States?
7. Who was the 6th president of the United States?
8. Who was the sixth president of America?
9. Who was the president of America in 1817?
10. Who was the Secretary of State under James Madison?
11. Who is the father of the constitution in America?
12. Who was the founder of the constitution in America?
13. Who is the vice president who is known best for being the namesake of gerrymandering?
14. Who was the Vice President under James Madison?
15. Who is Elbridge Thomas Gerry?
16. Who is the president who act a significant role to stop the war of 1812?
17. Who was the wife of the president John Adams?
18. Who is current President of the United States?
19. Who is the first African American President?
20. Who was the president who born in Honolulu, Hawaii?
21. Who is Barack Hussein Obama?
22. Who is the American retired professional tennis player who has been a World No. 1?
23. Who is Andre Kirk Agassi?
24. Who was the dominant players from the early nineteenth?
25. Who was the first male player to win all four Grand Slams?
26. Who is the player who is called "The Punisher"?
27. Who is the top ranked Swiss tennis player?
28. Who won 17 Grand Slam singles titles?
29. Who have succeed 17 Grand Slam singles titles?
30. Who has the most titles at the Australian Open in addition to Federer and Agassi?
31. Who has the most titles at Wimbledon in addition to Federer?
32. Who has the most titles the US Open in addition to Federer and Pete Sampras?
33. Who has won the most matches in men's Grand Slam tournaments?
34. Who is Rafael Nadal?
35. Who is the most known Spanish tennis player?
36. Who is the player who is called the "King of Clay"?
37. Who is the tennis player who was born on 29 April 1970?
38. Who is the tennis player who was ranked as the World No 1 player in 1999?
39. Who is the best tennis player in 1999?
40. Who won 17 ATP Masters Series titles?
41. Who is the tennis player who was ranked as the World No 5 player by the ATP in 2013?
42. Who is the tennis player who have reached the Wimbledon final eight times?
43. Who was the scientist who conceived the electro-mechanical device?
44. Who was the scientist who conceptualise the electro-mechanical device?
45. Who was in the team of Aiken in IBM?
46. Who invented Mark II?
47. Who is the founder of the Harvard Computation Laboratory?
48. Who joined Naval Reserve in 1943?
49. Who was the first one who use the term 'Bug' for a computer fault?
50. Who was the scientist who born in New York in 1906?
51. Who was the first person to 'debug' a computer?
52. Who invented the first English-language data processing compiler?
53. Who was the "Man of the Year" in computer science in 1969?
54. Who won the the National Medal of Technology in 1991?
55. Who was the scientist who born in Milan in 1847?
56. Who was the last child of Samuel and Nancy Edison?
57. Who is the mother of Thomas Edison?
58. Who is the scientist who attended school only for a few months?
59. Who is the scientist who was called 'addled' in the school?
60. Who was the inventor who said "My mother was the making of me"?
61. Who taught Thomas Edison in his childhood?
62. Who invented coke smelting in 1709?
63. Who founded the world's first metallurgy laboratory?
64. Who received a patent for his sand casting in 1708?
65. Who was the Ancient Greek philosopher who talked about the art of wealth acquisition?
66. Who is the famous economist in the medieval times?
67. Who is the the "Father of Modern Economics"?
68. Who wrote "The Wealth of Nations"?
69. Who is one of the classical economists who heir Smith?
70. Who objurgate the capitalist system?
71. Who depict the the capitalist system as estrange?
72. Who was against the absence of government from economic matters in twentieth century?
73. Who discouraged using socialist economy?
74. Who concenter on accomplishing better pecuniary strategies instade of socialist economy?
75. Who tried to provide neoclassical microeconomic mechanisms to help analyze macroeconomic issues?
76. Who is one of the information economists?
77. Who is one of the development economists?
78. Who developed a theoretical computational model?
79. Who suggested the idea of measuring time and space of algorithms as a function of the length of the input?
80. Who propose the thought of evaluate the algorithms complexity?
81. Who propose the concept of NP-hard problems?
82. Who propose the idea of Quantum computers in 1982?
83. Who designed a quantum algorithm to solve factoring integers problem in 1994?
84. Who solve an NP-hard problem in polynomial time on a quantum computer?
85. Who designed a quantum algorithm in 1996?
86. Who is the Swedish chemist who invented dynamite?
87. Who invented dynamite?
88. Who is Alfred's brother?
89. Who is called by a French newspaper as the "merchant of death"?
90. Who is the mother of Alfred Nobel?
91. Who is the father of Alfred Nobel?
92. Who works as an architect and builder in Nobel's family?
93. Who received a Swedish patent for his percussion detonator in October 1863?
94. Who is the brother of Alfred who was killed because of the blew up of Alfred's factory?
95. Who is the Alfred's brother who was born after their family had moved to Russia?
96. Who won the Nobel Prizes in Chemistry in 1901?
97. Who won the Nobel Prizes in Physics in 1901?
98. Who won the Nobel Prizes in Physiology or Medicine in 1901?
99. Who won the Nobel Prizes in Literature in 1901?
100. Who won the Nobel Prizes in Peace in 1901?
B.2 Evaluation Questions for Huge Data

1. Who was beaten by Andy Murray in semi finals of 2013 Australian Open?
2. Who faced Murray on the Wimbledon center court?
3. Who is the Swiss professional tennis player?
4. Who is “The Punisher”?
5. Who appeared in 24 men’s Grand Slam finals?
6. Who is the Spanish professional tennis player?
7. Who is known for off-court impersonations?
8. Who served a career high of 42 aces in a match?
9. Who is the Scottish professional tennis player?
10. Who was born in Las Vegas?
11. Who was born in Omaha?
12. Who is the sixth president of United States?
13. Who served as the 42nd president of United States?
14. Who is the current president of United States?
15. Who is George Bush?
16. Who is the second president of United States?
17. Who is John Kennedy?
18. Who was born in New Heaven?
19. Who was an American statesmen and diplomat?
20. Who is the 44th president of United States?