



OPEN DOMAIN FACTOID QUESTION ANSWERING SYSTEM

FATİH ÖZKAN

SEPTEMBER 2015

OPEN DOMAIN FACTOID QUESTION ANSWERING SYSTEM

**A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES OF
ÇANKAYA UNIVERSITY**

**BY
FATİH ÖZKAN**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF
COMPUTER ENGINEERING**

SEPTEMBER 2015

Title of the Thesis: **Open Domain Factoid Question Answering System**


Submitted by **Fatih ÖZKAN**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.



Prof. Dr. Halil Tanyer EYYUBOĞLU
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Master of Science**.



Prof. Dr. Müslim BOZYİĞİT
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.



Assist. Prof. Dr. Murat SARAN
Supervisor

Examination Date: 16.09.2015

Examining Committee Members

Assist. Prof. Dr. Abdül Kadir GÖRÜR (Çankaya Univ.)

Assist. Prof. Dr. Murat SARAN (Çankaya Univ.)

Assist. Prof. Dr. Gönenç ERCAN (Hacettepe Univ.)








STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Fatih ÖZKAN
Signature : 
Date : 16.09.2015

ABSTRACT

OPEN DOMAIN FACTOID QUESTION ANSWERING SYSTEM

ÖZKAN, Fatih

M.Sc., Department of Computer Engineering

Supervisor: Assist. Prof. Dr. Murat SARAN

September 2015, 54 pages

Mankind has recently produced a vast amount of data. According to IBM, every day we create 2.5 quintillion bytes of data [1]. To find the relevant information in this huge body of data, we use search engines. These search engines locate the relevant data according to a user's query and return a list of results to the user, after which the user finds what he is seeking in this list. However, users may not wish to read a list of results and may prefer short, relevant answers to their query. For this reason, there is much research on Question Answering Systems. The task of these systems is the acquisition of relevant information to questions asked by users.

In this thesis, we developed a question answering system called *Prime* and describe how it works. *Prime* is a factoid question answering system. The term *factoid* is defined as “an item of unreliable information that is reported and repeated so often that it becomes accepted as fact.” Therefore, it gives only an answer to a user's factoid question. We also describe how we extract question triples (subject-predicate-object) based on dependency graphs and build SPARQL queries to query DBpedia

based on extracted triples in addition to describing how we map entities in questions to DBpedia entities.

We evaluated our system with QALD-5 Question answering challenge questions and show that our system promises worthy results. In the QALD-5 training test questions after excluding any unnecessary questions, our system answered 43% of the questions. In the test questions, our system answered 41% of the questions. Furthermore, we describe the evaluation results and evaluation metrics and compared our system with other QALD-5 participant systems.

Keywords: Natural Language Processing, Question Answering Systems, Semantic Web, DBpedia

ÖZ

AÇIK ALAN TEKİL YANITLI SORU CEVAPLAMA SİSTEMİ

ÖZKAN, Fatih

Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı

Tez Yöneticisi: Dr. Murat SARAN

Eylül 2015, 54 sayfa

İnsanoğlu son zamanlarda çok fazla veri üretti ve IBM'e göre hergün 2.5 kentilyon byte[1] veri üretiyoruz. Bu veriler içinde ilgili bilgileri bulmak için arama motorlarını kullanıyoruz. Bu arama motorları ilgili sonuçları kullanıcıların sorgularına göre bulup kullanıcılara sonuçları liste olarak göstermektedir bundan sonra ise kullanıcılar istedikleri bilgileri bu liste içerisinde kendilerinin bulması gerekmektedir. Fakat kullanıcılar, bu sonuç listesini okumak istemeyebilir ve sorgularına kısa cevaplar isteyebilirler. Bu nedenle Soru Cevaplama Sistemleri üzerinden çok fazla araştırma yapılmaktadır. Bu sistemlerin görevi kullanıcılar tarafından sorulan sorulara ilgili cevapları vermektir.

Bu tezde, Prime olarak adlandırılan bir soru cevaplama sistemi geliştirdik ve sistemin nasıl çalıştığını anlattık. Prime tekil yanıtli soru cevaplama sistemidir, bu nedenle sadece kullanıcıların tekil yanıtli cevap isteyen sorularına cevap verir.

Ayrıca, bağımlılık grafiklerinden nasıl soru üçlüsü (özne-yüklem-obje) çıkartılacağını, DBpedia'yı sorgulamak için soru üçlülerinden nasıl SPARQL sorguları oluşturulacağını, ve sorular içerisindeki varlıkları nasıl DBpedia varlıkları ile ilişkilendirdiğimizi açıkladık.

Sistemimizi QALD-5 soru cevaplama sistemleri yarışması sorularıyla test ettik ve sistemimizin iyi sonuçlar verdiğini gösterdik. QALD-5 eğitim setindeki sorulardan gereksiz soruları çıkarttıktan sonra sistemimiz %43 oranında soruları doğru cevapladı. Test setindeki sorularda ise sistemimiz %41 oranında soruları doğru olarak cevapladı. Ayrıca test sonuçlarını ve ölçütlerini açıkladık ve sistemimizin sonuçlarını diğer QALD-5 katılımcılarının sonuçları ile karşılaştırdık.

Anahtar Kelimeler: Doğal Dil İşleme, Soru Cevaplama Sistemleri, Sematik Web, DBpedia

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Assist. Prof. Dr. Murat SARAN for his supervision, special guidance, suggestions and encouragement through the development of this thesis.

I would like to thank my family to their continued support.

I would like to thank my wife, Zeynep, for her support, patience, help and understanding.

TABLE OF CONTENTS

STATEMENT OF NON PLAGIARISM.....	iii
ABSTRACT.....	iv
ÖZ.....	vi
ACKNOWLEDGEMENTS.....	viii
TABLE OF CONTENTS.....	ix
LIST OF FIGURES.....	xii
LIST OF TABLES.....	xiii
LIST OF ABBREVIATIONS.....	xiv
CHAPTERS:	
1. INTRODUCTION.....	1
1.1. Overview.....	1
1.2. Objectives.....	3
1.3. Organization of the Thesis.....	3
2. BACKGROUND.....	4
2.1. Question Answering System Types.....	4
2.1.1. Closed Domain.....	4
2.1.2. Open Domain.....	4
2.2. Question Types.....	5
2.2.1. Factoid Questions.....	5
2.2.2. List Questions.....	5
2.2.3. Definition Questions.....	6
2.2.4. Complex Questions.....	6
2.2.5. Speculative Questions.....	7
2.3. Semantic Web.....	7
2.4. Ontologies.....	9
2.5. Linked Data.....	12
2.6. Wikipedia.....	14
2.7. DBpedia.....	15

2.8.	Tokenization.....	15
2.9.	Part of Speech Tagging.....	16
2.10.	Named Entity Recognition.....	17
2.11.	WordNet.....	17
2.12.	SPARQL.....	18
2.13.	Lemon Model.....	19
3.	PREVIOUS RELATED WORKS.....	20
3.1.	Baseball.....	20
3.2.	Parry.....	21
3.3.	Lunar.....	21
3.4.	Qualm.....	21
3.5.	Start.....	21
3.6.	Wolfram Alpha.....	22
3.7.	IBM Watson.....	22
3.8.	AquaLog.....	22
3.9.	Pythia.....	23
3.10.	FREyA.....	23
4.	PRIME QUESTION ANSWERING SYSTEM.....	24
4.1.	Architecture of Prime and Prime Pipeline.....	24
4.2.	Application Development Environment.....	26
4.3.	Question Analysis.....	27
4.3.1.	Tokenization, Pos Tagging, Name Entity Recognition.....	27
4.3.2.	Question Classification and Answer Type Finding.....	27
4.3.2.1.	Rule Base Approach.....	30
4.3.2.1.1.	Advantages.....	30
4.3.2.1.2.	Disadvantages.....	30
4.3.2.2.	Machine Learning Base Approach.....	31

4.3.3.	Triple Extraction.....	32
4.4.	Entity Mapping.....	37
4.4.1.	Finding Properties and Ontologies.....	37
4.4.2.	Finding Name Entities.....	39
4.5.	SPARQL Generation.....	40
4.6.	Query Execution.....	42
5.	EVALUATION.....	44
5.1.	Testing Prime over QALD-5 Questions.....	44
5.2.	Test Results of the System.....	46
5.3.	Performance of the System.....	50
6.	CONCLUSION AND FUTURE WORK.....	52
	REFERENCES.....	R1
	APPENDICES.....	
A.	QALD-5 TRAINING AND TEST QUESTIONS.....	A1
B.	CURRICULUM VITAE.....	B1

LIST OF FIGURES

FIGURES

Figure 1	Google Search Results	2
Figure 2	Wolfram Alpha Search Result	2
Figure 3	Semantic Web Architecture.....	8
Figure 4	DBpedia Class Hierarchy.....	10
Figure 5	Linked Data Cloud.....	13
Figure 6	Prime Web Interface.....	25
Figure 7	Architecture of Prime	25
Figure 8	Sample Triple About Question Microsoft is location in Redmond.....	32
Figure 9	Sample Triple about Question Google has 49,829 employees...	33
Figure 10	Sample Triple about Question Albert Einstein was born in 14 March 1879.....	33
Figure 11	Sample Dependency Graph.....	34
Figure 12	Sample Parse Tree	34
Figure 13	Dependency Graph of Question “Which languages are spoken in Estonia?”.....	36
Figure 14	SPARQL Query to find Name Entities.....	39
Figure 15	SPARQL Query for Question “What is the currency of Turkey?”.....	41
Figure 16	SPARQL Query for Getting Additional Information of a DBpedia Resource	43

LIST OF TABLES

TABLES

Table 1	Sample Factoid Questions	5
Table 2	Sample List Questions	6
Table 3	Sample Definition Questions	6
Table 4	Sample Complex Questions.....	7
Table 5	Sample Speculative Questions.....	7
Table 6	Li and Roth’s Taxonomies (Question Classes).....	28
Table 7	Some of the Qtargets (Question Classes) of Hermjakob.....	29
Table 8	Moldovan Hierarchical Taxonomy.....	29
Table 9	List of Literal Types in SPARQL.....	42
Table 10	Prime’s Evaluation Results of QALD-5 Test And Train Question Set.....	47
Table 11	QALD-5 Evaluation Results.....	47
Table 12	Prime's Performance Metrics.....	51

LIST OF ABBREVIATIONS

QA	Question Answering
QAS	Question Answering System
NLP	Natural Language Processing
SCNLP	Stanford CoreNLP
HTML	HyperText Markup Language
RDF	Resource Description Framework
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
QALD	Question Answering over Linked Data
SPARQL	SPARQL Protocol and RDF Query Language
HTTP	HyperText Transfer Protocol
XML	Extensible Markup Language
OWL	Web Ontology Language
POS	Part of Speech
NER	Named Entity Recognition
YAGO	Yet Another Great Ontology

CHAPTER 1

INTRODUCTION

1.1. Overview

A Question Answering (QA) system provides specific answers to questions asked in natural language. For example, a user wants to learn about the 2014 World Cup winner and asks the question “Who won the 2014 World Cup?” A QA system will respond to this question with a specific answer “Germany.” Question answering systems are distinguished from traditional search engines by giving specific results to questions. Examples of traditional search engines include Google¹, Bing², Yahoo!³, and Yandex,⁴ and examples of QA systems include Wolfram Alpha⁵, Evi⁶, and Start⁷.

Traditional search engines work differently from question answering systems. Search engines only return a list of documents (as shown in Figure 1) or they return data that may comprise one document or thousands. Search engines or information retrieval systems do not return answers to the questions of the user, in spite of the fact that users may want specific answers. Therefore, users have to search for the answers that they are seeking in these documents. The answer that a user is seeking may be found anywhere in a document (web page) or a document may not contain the answer or the document may have changed.

¹ <http://www.google.com>

² <http://www.bing.com/>

³ <https://www.yahoo.com/>

⁴ <https://www.yandex.com/>

⁵ <http://www.wolframalpha.com>

⁶ <https://www.evi.com/>

⁷ <http://start.csail.mit.edu/index.php>

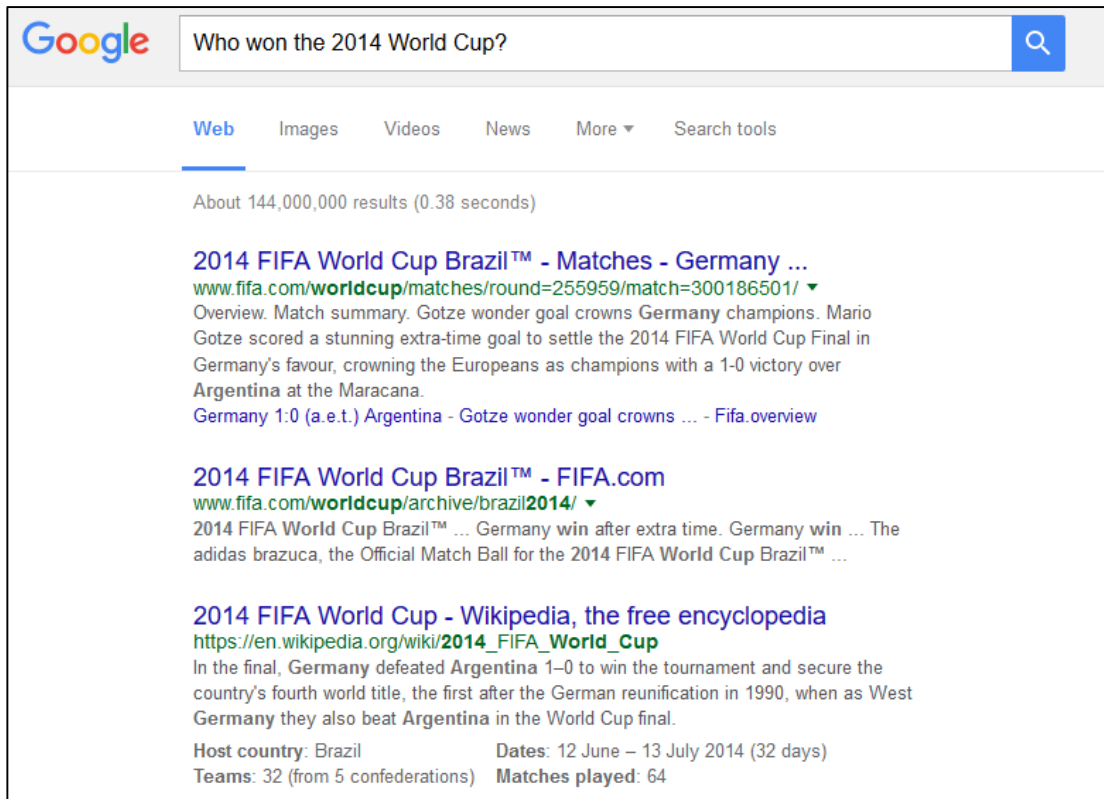


Figure 1 Google Search Results

With the question answering system, users do not have to search for answers in documents. Because these systems are a combination of Information Retrieval (IR), Information Extraction (IE) and Natural Language Processing, users can obtain specific answers to their questions, as shown in Figure 2.

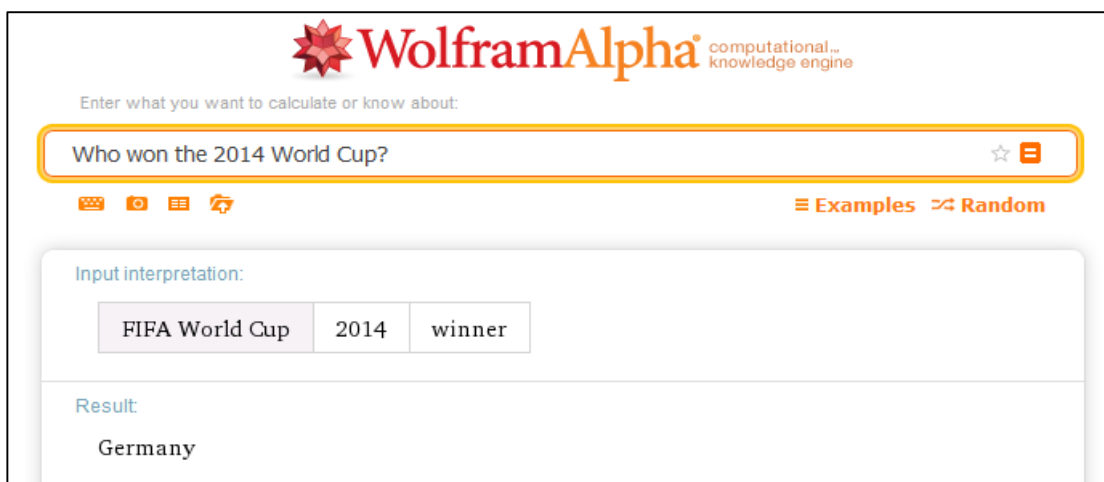


Figure 2 Wolfram Alpha Search Result

1.2. Objectives

The aim of this study is the design and development of a factoid question answering system, which can answer simple factoid questions. The system uses DBpedia as its data store. During this study, we explain how we extract triples from questions, how we map triple elements onto DBpedia ontology, and how we build SPARQL queries.

We aimed to develop a question answering system that:

- is able to answer a user's simple factoid questions quickly and correctly;
- creates a question answering system that works with different triple store data sources such as MusicBrainz, DBpedia, and GeoNames etc.; and
- has the ability to extend itself to answer List, Definition, and Complex questions.

1.3. Organization of the Thesis

This thesis contains six chapters. In Chapter 2, the background is presented. We describe the types of Question Answering systems, the Semantic Web, Ontologies, DBpedia, Wikipedia and NLP techniques. In Chapter 3, we present related works about Question Answering Systems. In Chapter 4, the general architecture of the prime question answering system is explained. In Chapter 5, evaluation methods and test results of the study are presented.

Chapter 6 includes the conclusion part.

CHAPTER 2

BACKGROUND

2.1. Question Answering System Types

There are two types of question answering systems: Closed domain and Open domain.

2.1.1. Closed Domain

Closed domain question answering systems deal with specific domains, such as medicine, music, weather forecasting, sports, etc. Early question answering systems starting from the 1960s are closed domain systems. The first domain specific QA system was BASEBALL [2], which was designed to answer questions about the American League for one year. Closed domain systems use pre-tagged corpora or knowledge bases.

2.1.2. Open Domain

After the invention of the World Wide Web, many kinds of data became public and finding a broad range of information became easy. Open domain question answering systems deal with nearly everything. These systems find answers from web documents, Wikipedia, structured databases or semi-structured databases, such as DBpedia.

2.2. Question Types

There are several question types in QA systems. These question types include:

- Factoid Questions
- List Questions
- Definition Questions
- Complex Questions
- Speculative Questions

2.2.1. Factoid Questions

Factoid questions have one correct answer and this answer may be a person, date, location etc. Answers to factoid questions can be extracted from texts. Factoid questions are easy to understand and answer, so this type of question is the simplest form of question [3].

“What is the capital of Turkey?” is a factoid question the answer to which is “Ankara.”

Table 1 Sample Factoid Questions

Question	Answer
What is the population of Japan?	126,434,964
When was Apple founded?	1976-04-01
Who was the first Nobel Prize winner?	Wilhelm Conrad Röntgen
Where is New York?	United States of America
Who directed Avatar?	James Cameron

2.2.2. List Questions

A list question yields a *list of items* as its answer. Answering this type of question is more difficult than answering factoid questions as a list of items in the answer can be a different document. The systems may necessitate the scanning of thousands of documents.

Table 2 Sample List Questions

Question	Answer
Actors of The Matrix?	Keanu Reeves, Laurence Fishburne, Carrie-Anne Moss, Hugo Weaving ...
What cities are in Turkey?	Ankara, Istanbul, Izmir, Antalya ...
Name of all the players in the Premier League?	Ramires, Rolando Aarons, Fabricio Coloccini, Mesut Özil...
Give me all cosmonauts	Mikhail Korniyenko, Anatoli Ivanishin, Sergei Revin...

2.2.3. Definition Questions

The answer to definition questions is a merger of results of simple phrases from different documents. Answers are collected from different sources and relevant phrases are extracted from those documents. After phrase extraction, the system must reorder and merge the phrases in order to generate a meaningful answer. Answering this type of question requires systems that are more sophisticated.

Table 3 Sample Definition Questions

Question	Answer
Why is sky blue?	We see a blue sky, because of the way the atmosphere interacts with sunlight.
Define “Eternal Life”	Eternal life traditionally refers to continued life after death
What does “obfuscate” mean?	Make obscure, unclear, or unintelligible.

2.2.4. Complex Questions

Complex questions contain several sub-questions inside the main question. For example, the question “Which actors born in Melbourne also won the Academy Awards?” has two sub-questions:

- “Which actors were born in Melbourne?” and
- “Which actors won the Academy Awards?”

In order to answer this question, we have to answer the sub-questions individually and intersect the answers to extract the final answer(s).

Table 4 Sample Complex Questions

Question	Answer
Of all European countries, which has the smallest area?	Vatican City
What is the population of the capital of France?	2,273,305
Who is the Formula 1 race driver with the most races?	Rubens Barrichello

2.2.5. Speculative Questions

Speculative questions are difficult to answer. To answer this type of question, QA systems have to use reasoning techniques. The answer to a speculative question may be in a multiple document or data store. The QA system has to find clues in the documents in order to answer a speculative question. After finding these clues, the system uses reasoning techniques to find the correct answer(s).

Table 5 Sample Speculative Questions

Question
Will Microsoft buy Facebook?
How can the US capture Edward Snowden?
Will Galatasaray be champion of the Super League of this year?

2.3. The Semantic Web

The Semantic Web is an extension of the Web through standards by the World Wide Web Consortium (W3C). In 2001, Tim Berners-Lee, who invented the World Wide Web (WWW) in the late 1980s, James Hendler and Ora Lassila [4] wrote an article about the Semantic Web, in which they explain with this sentence: “A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities.” Tim Berners-Lee also said, “The Semantic Web is not a separate Web, but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation,” which means the Semantic Web is the upper level of today’s Web.

The web content is presented in a way that people can understand. The web content contains text, images, links, sounds and videos. This content can be viewed with the help of a browser. Humans are able to read and understand web content easily; however, machines cannot understand text in the same manner as humans. The aim of the Semantic Web is to “link the web pages and applications with each other, share data between web pages and applications and make the content more *machine-accessible*.”

To search today’s Web content, we use keyword-based search engines such as *Google, Yahoo!* and *Bing*. These search engines, as we stated in the introduction, give users a list of page links. The user who makes the search has to select the relevant page(s) to find what he is seeking. Therefore, the term *information retrieval*, used in association with search engines, is somewhat misleading; *location finder* might be a more appropriate term. [5] Machines are required to interpret the content and extract any meaningful information to give to users. However, machines do not perform as well as humans with regard to interpreting content and extracting meaningful information.

As stated earlier, the real meaning of the Semantic Web is “Making web content *readable* and *accessible* by humans and machines.” Figure 3 illustrates the Architecture of the Semantic Web.

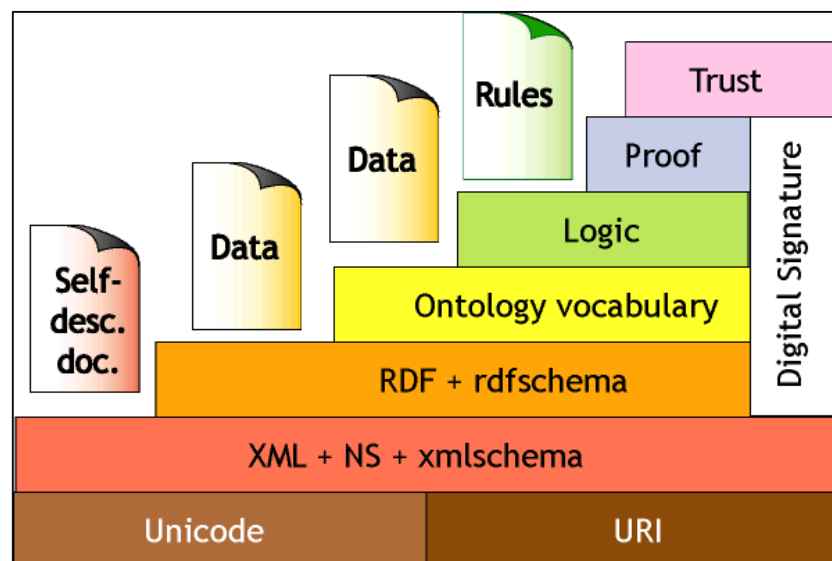


Figure 3 Semantic Web Architecture

The layers of Semantic Web [5] are briefly described below.

- Unicode and URI
 - Unicode is an international standard character set. With Unicode characters, the Semantic Web communicates in any language.
 - A URI is the unique name of things and concepts on the Web.
- XML
 - XML (Extensible Markup Language) is markup language used to create documents for humans and machines. With XML data, not only humans but also machines can understand data.
- RDF
 - RDF is a set of data model resources and the relations between them. It contains three object types: *resources*, *properties* and *statements*. RDF uses URI to identify resources and items, XML for serializing data to machine readable format.
- Ontology
 - Ontology is the shared common understanding of a domain. A more detailed explanation is found in Section 2.4 Ontologies.
- Logic
 - A logic layer is provided to create and write application-specific rules.
- Proof
 - A proof layer executes the rules that are created in the logic layer, and evaluates them using the trust layer.
- Trust
 - The trust layer is the last layer of Semantic Web architecture and ensures trustworthiness of data by using *digital signatures*.

2.4. Ontologies

The meaning of ontology [6] is the philosophical study of the nature of being, becoming, existence or reality, as well as the basic categories of being and their relations.

Semantic Web ontologies play an important role. Today the World Wide Web consists of billions of web pages used by billions of people. Therefore, the exchange

of information is a difficult task. Ontologies provide a common model for the exchange of information.

Ontology is a basic shared common understanding of a domain. These may well be a basic subject for humans, but it is important for machines. Without these specifications, machines cannot understand the differences in meaning or usage. For example, a web site can use *car* as a text whereas another web site may use *automobile*, or one application uses the term *zip* while another uses *code*. The terms in these examples are identical for humans but are considered different things by machines. Ontologies map these terms.

The structure of ontology is based on four components.

- Classes (concepts or terms) are abstract groups, sets or collections of objects. Classes have hierarchical structures between them, namely *super-class* and *sub-class*. Some classes in DBpedia ontology are shown in Figure 4. All classes are derived from the class *Thing*; this class is a super-class of all classes. In the figure, there are three *sub-classes*: *Agent*, *Event*, and *Place*. The agent class has a sub-class *Person* and the Person class has a sub-class *Artist*. This hierarchical structure can be of infinite depth.

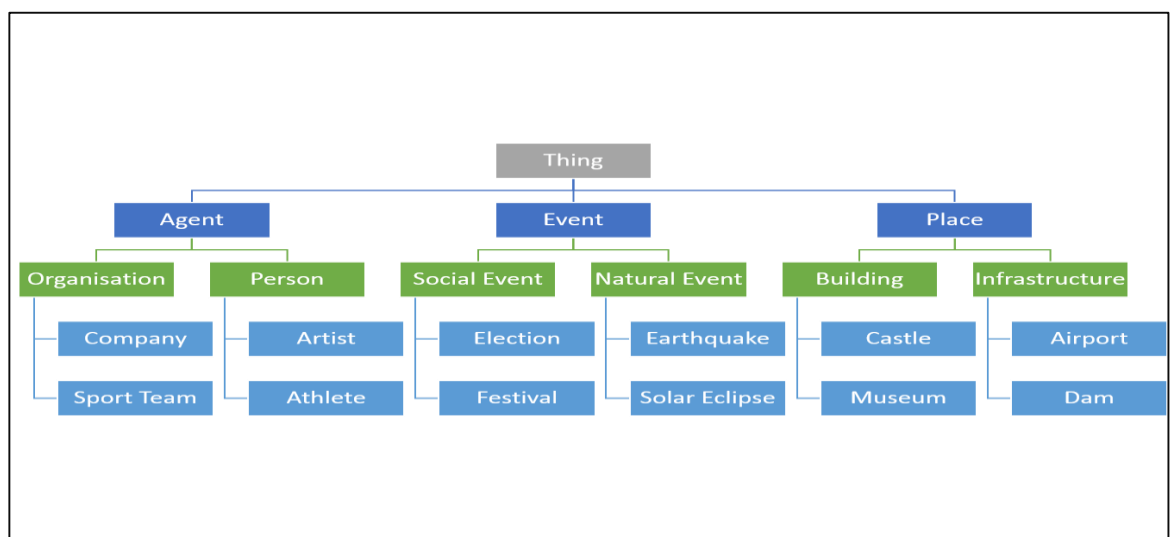


Figure 4 DBpedia Class Hierarchy

- Instances are basic “ground level” components of an ontology. For example, the Empire State Building could be an instance of *Building*, or *Place*.
- Taxonomy is the hierarchical representation of classes.
- Attributes (properties) describe objects (instances or classes) in the ontology. For example, the class *Person* has an attribute *hasName*, *hasSurname* and *hasAge*.
- Relationships express how objects in the ontology are related to each other. For example, person *born-in* Ankara. The *born-in* relationship tells us where each person is born.
- Axioms are knowledge definitions in the ontology that have been explicitly defined and that have not been proven true. For example, the Empire State Building is an instance of *building*; it is a place.

Natalya F. Noy and Deborah L. McGuinness explain: “Why do we need ontology?” [7]

- Sharing common understanding of the structure of information among people or software agents;
- Enabling reuse of domain knowledge making explicit domain assumptions;
- Separating domain knowledge from operational knowledge;
- Analyzing domain knowledge.

W3C provides several ontology languages for the Web:

- RDF is a set of data model resources and the relations between them. It contains three object types: *resources*, *properties*, and *statements*. RDF is based on XML.
- RDF Schema is a vocabulary description language for describing properties and classes of RDF resources, with a semantics of generalization hierarchies of such properties and classes.
- OWL is a richer vocabulary description language for describing properties and classes, such as relations between classes (e.g., disjointness), cardinality (e.g., “exactly one”), equality, richer typing of properties, characteristics of properties (e.g., symmetry) and enumerated classes.

2.5. Linked Data

Linked Data pertains to using the Web to connect related data that was not previously linked, or it may pertain to using the Web to lower the barriers to linking data currently linked using other methods. [8]

Web documents are built with HTTP that contains hyperlinks (URLs) to connect to other documents. Linked data, which uses HTTP and URI web technologies, extends the current Web to share information between machines. It relies on document data in RDF format. This enables different data stores to share information with each other and query related data. Linked data connect different data stores, as shown in Figure 5.

Identical entities in datasets connect to each other with specific vocabularies, such as *owl:sameAs*. With this vocabulary, we can easily locate the same entity in other datasets.

Linked datasets are growing day by day and these datasets share information between each other. DBpedia is the largest dataset in the linked data cloud and it is in the heart of the cloud. There are many datasets in different domains, such as geographic, medicine, media, linguistics, government, publications, life science etc.

Berners Lee [10] outlined the following rules for publishing linked data on the Web. The principles of Linked Data:

1. Use URIs as a name of things;
2. Use HTTP URIs so that people can look up those names;
3. Provide useful information using the standards (RDF, SPARQL) when someone looks up a URI; and
4. Include links to other URIs so that they can discover new things.

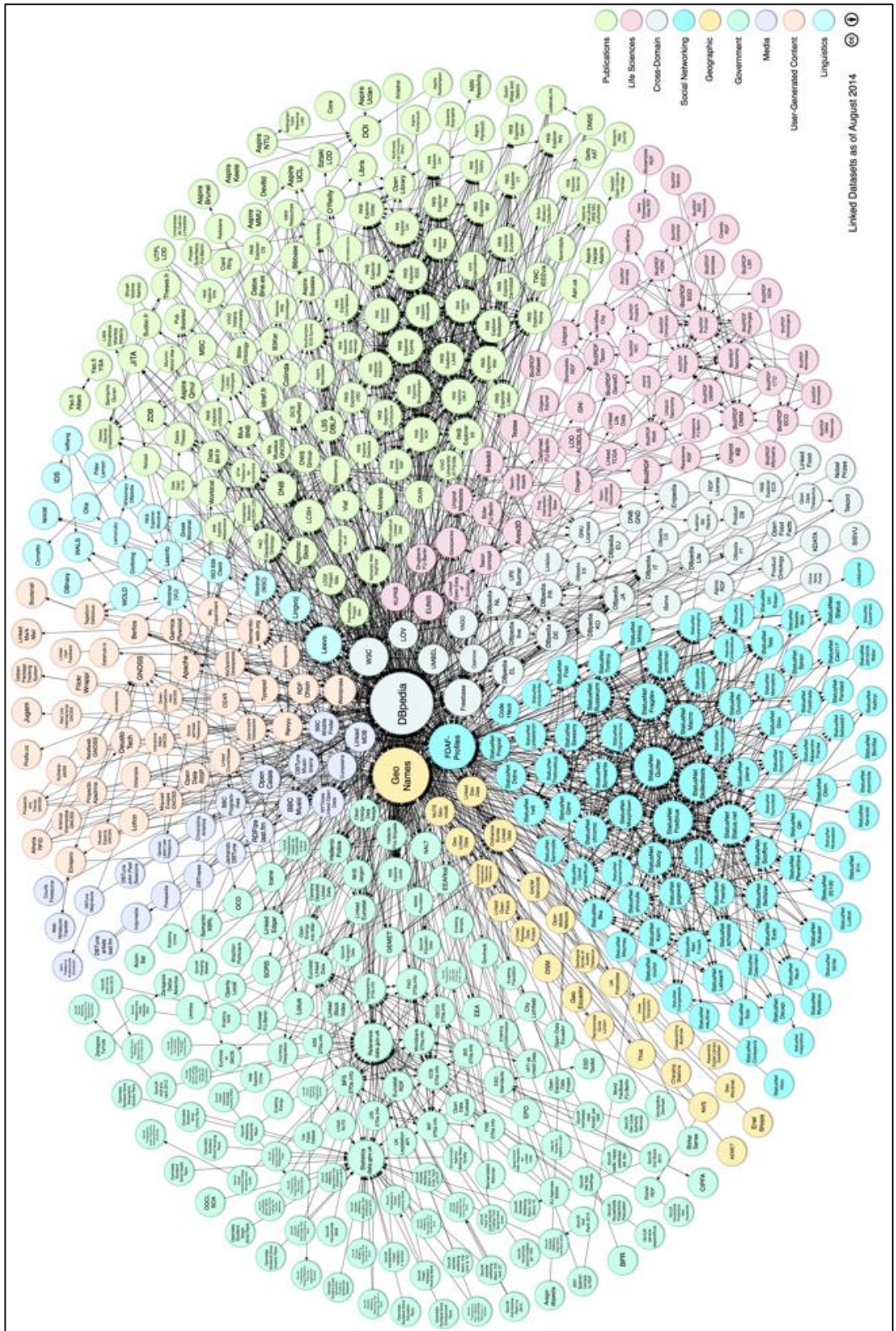


Figure 5 Linked Data Cloud [9]

2.6. Wikipedia

Wikipedia is a free encyclopedia that has been developed through community effort. Anyone can create and edit articles on Wikipedia. Wikipedia uses special markup language especially developed for wikis. Output of the WikiMarkup is html. Wikipedia articles consist of links, texts, images, tables and links in the article text that direct to external web pages or other Wikipedia pages.

Wikipedia pages are an unstructured form of data. To extract data from these articles is a very challenging task.

Wikipedia has semi-structured tables in articles called an *InfoBox*. These *infoboxes* hold the summaries of articles and key information about pages. For example, the infobox for the Wikipedia page about Turkey has *Capital, Largest City, Official Language, Government, Currency, Time Zone*, etc. information. This information is semi-structured and can be easy extract. Not every infobox on a Wikipedia page contains the same information. As already stated, Turkey's infobox has information about location, but Michael Jackson's infobox has *Birth Date, Death Date, Religion, Parents, Occupation*, etc.

Wikipedia is heavily used in NLP tasks, search engines and knowledge miners. Information in Wikipedia is unstructured, whereas DBpedia is an extraction of structured information from Wikipedia articles and the building of a linked data resource from it. The structured information used to build DBpedia includes infoboxes, links and tables.

Some statistics of the English Wikipedia by the year 2015:

- 4,941,064 articles
- 36,926,501 pages
- 863,438 files
- 784,337,633 edits
- 25,914,843 users

2.7. DBpedia

Wikipedia has grown into one of the central sources of knowledge for humanity. In reality, it is the largest encyclopedia on the Web and is maintained by thousands of contributors. Wikipedia is available in more than 285 languages. The English Wikipedia contains more than 3.5 million articles. The DBpedia project [11] aims to extract information from Wikipedia and make this information available on the emerging web of data.

The three major parts of the DBpedia project are the website, the datasets and the Extraction Framework. The dataset based on English Wikipedia infoboxes is probably the most interesting. Furthermore, DBpedia has 125 localized editions (French, German and Spanish, etc.) The extraction work is performed by the DBpedia Extraction Framework. The DBpedia ontology has been created in order to classify this extracted data. It is a cross-domain ontology based on infobox templates in Wikipedia articles. The ontology currently covers 359 classes, which form a subsumption hierarchy and are described by 1,775 different properties. In addition, the DBpedia 3.8 knowledge base describes by the year 2015:

- 4.58 million entities
- 1,445,000 people
- 735,000 places
- 229,000 creative works (123,000 music albums, 87,000 films, 19,000 video games)
- 241,000 organizations
- 251,000 species
- 6,000 diseases
- 25.2 million links to images, 29.8 million links to external web pages

2.8. Tokenization

Tokenization is a process to split sentences into words, phrases, and symbols. After tokenization, each word will be called a *token*. Basic tokenization splits words with a whitespace. This approach works with Latin, Cyrillic or Greek-based languages,

which are called segmented languages. For example, “What is the capital of Turkey?” would be split as [What] [is] [the] [capital] [of] [Turkey] [?]

However, in unsegmented languages such as Chinese, Japanese and Thai, this approach will not work. In Chinese, since there is no space between words or sentences, it is hard to define where a word starts and ends. For this reason, it is hard to tokenize Chinese sentences. Chinese characters are logograms [12], which represent a word. For example, the word “volcano” (火山) is in fact the combination of 火 (fire) and 山 (mountain). While tokenizing words in Chinese, tokenization must be done carefully. If a tokenizer splits these words without the context of the sentence, it changes the meaning of the sentence. There are various tokenizers for Chinese, one of which is the SCNLP toolkit.

In order to tokenize accurately, some language tasks have to be applied in the tokenizer.

- Segmenting sentences into tokens
- Handling abbreviations
- Handling hyphenated words
- Handling numerical and special expressions (Dates, times, measures etc.)

Tokenization is generally considered an easy task in NLP, but it is not easy as it appears. As stated previously, not all languages are the same and there are some tasks for each language.

2.9. Part of Speech Tagging

Part of speech (POS) tagging is a task to identify tags of each word after tokenization. POS tagging identifies nouns, verbs, adverbs, prepositions, etc. For example, “What is the capital of Turkey?” would correctly be tagged “What/WRB is/VBZ the/DT capital/NN of/IN Turkey/NNP ?/.” In this output, WRB stands for “wh-adverb”, VBZ for “verb, 3rd person singular present tense”, DT for “determiner”, NN for “noun, singular or mass” IN for “preposition/subordinating conjunction”, and NNP for “proper noun, singular”.

2.10. Named Entity Recognition

Named entity recognition is a task to identify entities in a text. The term Named Entity was first used at the Sixth Message Understanding Conference (MUC). [13] The basic entity types are *person*, *location*, *time*, *date*, *organization*, *quantity*, *money* and *percentage*. Named entity types can be extended by recognizer libraries. For example, *location* can be extended to *Country*, *City*, *Continent*, *Region*, *State*, etc.

Name entity recognition is not a simple task. Some of the challenges in an NER task include the following:

- It is difficult to recognize a named entity just by looking at its post tags.
- Some recognizers rely on capitalization. For example, if we write “*North America*,” the recognizer maps this phrase as *location*; if we write “*north America*,” the recognizer just maps America as a location.
- There is ambiguity in NER. An entity can be a location, person or organization.
- The data model used by a recognizer might not contain every domain. A recognizer will find named entities in trained domains.

2.11. WordNet

WordNet [14] is a huge, manually built lexicon of English. It contains nouns, verbs, adjectives, and adverbs that are grouped into sets of synonyms called synsets. The semantic and lexical relations, such as hypernym and hyponym, connect the WordNet synsets. These relations form a hierarchy of synsets and are very useful in the ontology building process. One of the important advantages of using the man-made WordNet hierarchy for ontology building is that it has a consistent taxonomy of concepts, which is a requirement in the ontology definition. WordNet includes the following semantic relations:

- Synonymy, which is WordNet’s basic relation as WordNet uses sets of synonyms (synsets) to represent word senses. Synonymy (*syn* same, *onyma* name) is a symmetric relation between word forms.

- Antonymy (opposing-name) is also a symmetric semantic relation between word forms, and is especially important in organizing the meanings of adjectives and adverbs.
- Hyponymy (sub-name) and its inverse, hypernymy (super-name), are transitive relations between synsets. Because there is usually only one hypernym, this semantic relation organizes the meanings of nouns into a hierarchical structure.
- Meronymy (part name) and its inverse, holonymy (whole name), are complex semantic relations. WordNet distinguishes component parts, substantive parts and member parts.
- Troponymy (manner name) is for verbs what hyponymy is for nouns, although the resulting hierarchies are much shallower.
- Entailment relations between verbs are also coded in WordNet.

2.12. SPARQL

SPARQL (pronounced “sparkle”, a recursive acronym for **SPARQL Protocol and RDF Query Language**) is an RDF language that is a semantic query language for databases and is able to store and retrieve data stored in a Resource Description Framework (RDF) format. It was made a standard by the RDF Data Access Working Group (DAWG) of the World Wide Web Consortium. It is recognized as one of the key technologies of the Semantic Web. On 15 January 2008, SPARQL 1.0 became an official W3C recommendation followed by SPARQL 1.1 in March 2013.

To query DBpedia, MusicBrainz, data.gov, DrugBank, FactForge etc., we have to build a valid SPARQL query from the question asked by the user.

The simple SPARQL query given below, which can provide an answer to the question “What is the currency of Turkey?”

```
PREFIX dbpedia: <http://dbpedia.org/resource/>  
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
```

```
SELECT ?currency  
WHERE { dbpedia:Turkey dbpedia-owl:currency ?currency .}
```

The lines with start *PREFIX* are namespaces. The variable named with *?currency* is the information we want to retrieve. *dbpedia:Turkey* is the subject, and *dbpedia-owl:currency* is the predicate.

SPARQL is similar to SQL (Structured Query Language). They have common keywords such as *SELECT*, *WHERE*, *FROM*, *GROUP BY*, *ORDER BY*, *DISTINCT* etc. There are specific keywords for SPARQL, such as *FILTER*, *OPTIONAL*, *PREFIX*, *CONSTRUCT*, *ASK* and *DESCRIBE*.

2.13. Lemon Model

Lemon (Lexicon Model for Ontologies) [15] is a model developed in the Monnet project to be a standard for sharing lexical information on the Semantic Web. The Lemon model can be used with different lexical resources. Some of the ontologies already implemented include FrameNet, OmegaWiki (German, English), VerbNet, Wiktionary.de, Wiktionary.en and WordNet. The aim of Lemon is to unify the formats of these models.

The Lemon-model has other advantages:

- Lemon is an RDF file, so we can query the model similarly to querying DBpedia with the Apache Jena Framework.
- It is not necessary to find every synonym for a word and query DBpedia.
- The Lemon model gives us every related DBpedia ontology class and property.

CHAPTER 3

PREVIOUS RELATED WORKS

In the early 1960s and 1970s, question answering systems started to develop. The first question answering system was BASEBALL [2]. BASEBALL was a domain-specific QA system that was designed to answer the American League over one season. Other QA systems include PARRY [16], Lunar [17] [18], Protosynthex [19], QUALM [20] [21] and PHLIQA1 [22]. These systems work with structured databases and text documents.

QA systems such as START [23] [24] were designed to work with semi-structured documents and web documents. START has its own knowledge base and with this knowledge base, users can access questions very easily. START analyzes English text, understands the question and transforms the question into a representation of a knowledge-based query.

3.1. BASEBALL

An early adaptor of QA systems was BASEBALL [2]. BASEBALL is a domain-specific QA system. It was designed to answer questions about the American League for one year. The program can answer questions such as “Who did the Red Sox lose to on July 5?” or “Did every team play at least once in each park in each month?” BASEBALL read questions from punched cards, followed by the program analyzing the question and then creating the query for the structured database. BASEBALL is very important because it was the first example of a program that used Natural Language Processing. It is very simple, and as previously stated, can only answer its own baseball database.

3.2. PARRY

PARRY [16] is a system developed by Colby for a contemporary of ELIZE. PARRY simulated a patient with paranoia that was demonstrated in a version of the Turing test. Psychiatrists were unable to determine whether PARRY was a person or a machine.

3.3. Lunar

The Lunar system [17] [18] was developed for scientists to access information on the lunar rocks that were collected during the Apollo moon mission. The Lunar system is a closed domain QA system. It analyzes a question and converts this question into a database query for querying information in the database about lunar rocks.

3.4. QUALM

The Question Answering Language Mechanism (QUALM) [20] [21] is a language independent question answering system. QUALM receives English input and responds with English, Spanish, Russian, Dutch or Chinese. QUALM is formed from four different NLP systems, namely SAM, PAM, COIL (Conceptual Objects for Inferencing in Language) and ASP (Answer Selection Program).

SAM and PAM are query-understanding systems. QUALM is the core program and SAM and PAM implemented by QUALM for question answering.

3.5. START

START (SynTactic Analysis using Reversible Transformation) [23] [24] was built at the MIT Artificial Intelligence Laboratory. START analyses a text and produces an answer from its own knowledge base. The knowledge base was created from unstructured Internet data. A user can retrieve information by querying in English followed by START responding in English.

3.6. Wolfram Alpha

Wolfram Alpha is a computational knowledge base that works by using its vast store of expert-level knowledge and algorithms to automatically answer questions, do analyses and generate reports. Wolfram Alpha uses thousands of knowledge bases for information. It also has its own primary knowledge base but uses other knowledge bases. Wolfram Alpha parses a query and relays the information to the user. It gives not only text responses but also maps, images, tables and equations. If a user asks a question such as “Where is Turkey?” Wolfram Alpha produces a map. If user asks a question such as “Turkey,” Wolfram Alpha responds by producing a large amount of information about the question such as *Name, Flag, Map, Geographic Information, Demographic Information, Cities* etc.

3.7. IBM Watson

Watson [25] is the latest and most well-known QA system ever developed. Watson beat two Jeopardy (a well-known TV show that has been on air for more than 25 years in America) champions in a real-time game on 14 January 2011. Watson uses not only NLP for its QA system; it also uses AI, IR, NLP, Machine Learning, Knowledge Representation and Reasoning. All of these components in Watson are called DeepQA.

3.8. AquaLog

AquaLog [26] is a question answering system based on Semantic Web architecture that takes natural language queries and ontology as an input and returns an answer. AquaLog uses the GATE NLP⁸ toolkit, string metrics algorithms, WordNet and ontology based similarity services for relationships and classes to generate queries. AquaLog translates questions asked in natural language into query triples (subject-predicate-object) and then translates those triples into ontology compatible triples.

⁸ <https://gate.ac.uk/>

3.9. Pythia

Pythia [27] is a question answering system that carries out a deep linguistic analysis on the question asked by the user. This approach enables Pythia to handle complex questions and queries. Pythia uses manually created lexicons for datasets and questions that make it domain-specific. It cannot answer questions if the lexicon was not designed for a specific domain.

3.10. FREyA

FREyA [28] is an interactive question answering system that uses enhancement methods such as feedback and clarification dialogs for improved precision and recall, thereby making it different from other question answering systems. FREyA uses the Stanford parser for Syntactic parsing and analysis and then uses several rules in order to identify Potential Ontology Concepts. After that, FREyA looks up Ontology concepts such as instances, classes, properties, etc. While searching ontology concepts, if FREyA encounters ambiguous concepts, it prompts the user with a disambiguation dialog. This dialog helps the user to identify ambiguities in the question. There is another dialog called the mapping dialog. If FREyA cannot map Ontology Concepts automatically onto Potential Ontology Concepts, this dialog helps the user to map Potential Ontology Concepts onto Ontology Concepts.

CHAPTER 4

PRIME QUESTION ANSWERING SYSTEM

In this study, we developed a question answering system called *Prime*. The details of Prime will be presented in this chapter.

4.1. Architecture of Prime and Prime Pipeline

Prime has modular architecture and supports multiple user interface, including a command-line and web interface. It also can be used as a library on Java platforms. In text-based command line interface, Prime takes a question as an input and prints the answer as an output. In Web interface, Prime takes a question as an input and shows a list of answers with names, descriptions, images and links to DBpedia⁹ and Wikipedia,¹⁰ as shown in Figure 6.

To answer a question, several stages are executed by every question answering system. These stages may differ from system to system, but the general architecture is always the same. In Prime, we developed a pipeline architecture called Prime Pipeline to execute these stages. This pipeline is described in Figure 7.

Prime Pipeline contains several stages: (1) Question Analysis, (2) Entity Mapping, (3) SPARQL Generation, and (4) Query Execution. In this chapter, each stage is discussed in detail.

⁹ <http://wiki.dbpedia.org/>

¹⁰ <https://www.wikipedia.org/>



Figure 6 Prime Web Interface

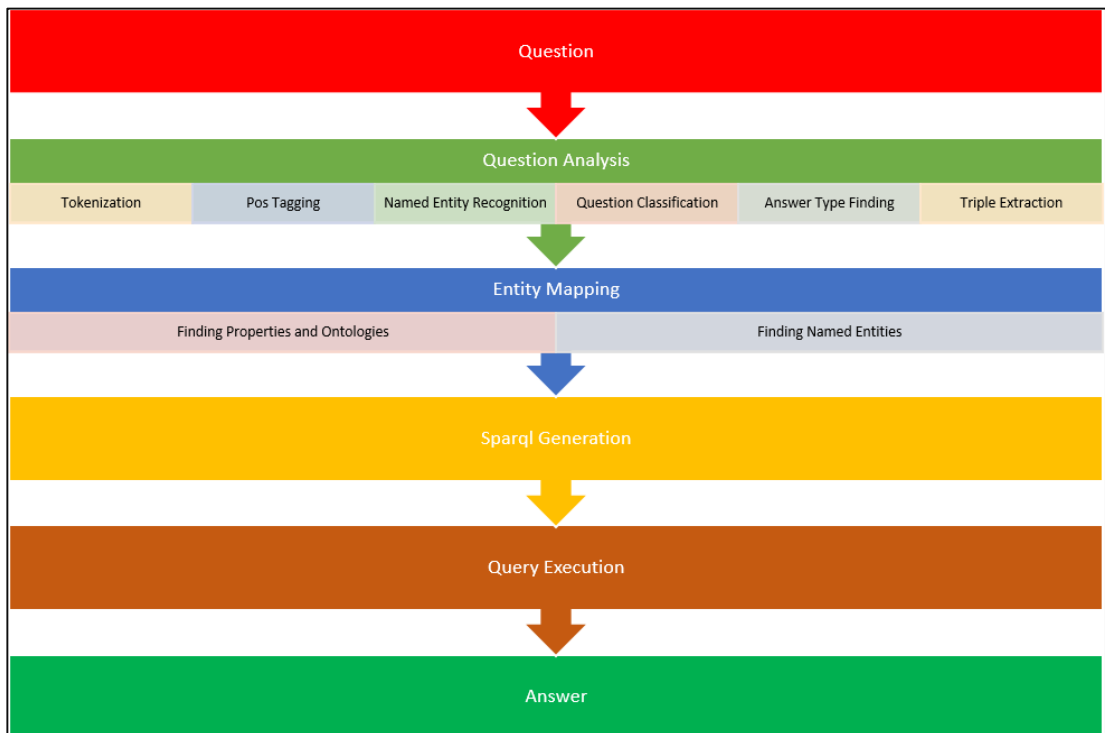


Figure 7 Architecture of Prime

4.2. Application Development Environment

Prime was developed with the Java programming language and uses external libraries, tools and technologies for various tasks. These tools, libraries and technologies include:

- the Java Programming language¹¹
- IntelliJ IDEA IDE¹²
 - We developed Prime in IntelliJ Idea IDE.
- Play Framework¹³
 - We use play framework for our web page.
- Stanford CoreNLP¹⁴ [29]
 - The natural language processing library contains several NLP tasks such as POS tagging, tokenization, NER etc.
- WordNet¹⁵ – an English Lexical Database
- The Lemon (Lexicon Model for Ontologies)-Model¹⁶
- the Jwnl WordNet Library¹⁷
 - Jwnl is a WordNet library written in Java. With this library, a text-based WordNet library can be queried.
- Apache Jena¹⁸
 - Semantic Web Framework
 - Jena is a semantic web framework. With Jena, we built SPARQL queries to query DBpedia.
- Open Ephyra's¹⁹[30] question classification module.

¹¹ <https://www.java.com/>

¹² <https://www.jetbrains.com/idea/>

¹³ <https://www.playframework.com/>

¹⁴ <http://nlp.stanford.edu/software/corenlp.shtml>

¹⁵ <https://wordnet.princeton.edu/>

¹⁶ <http://lemon-model.net/>

¹⁷ <http://sourceforge.net/projects/jwordnet/>

¹⁸ <https://jena.apache.org/>

¹⁹ <http://www.ephyra.info/>

4.3. Question Analysis

The first stage of processing in any Question Answering System is Question Analysis. This part of the system receives the unstructured text data and identifies any syntactic and semantic elements of the question. This module is the starting point of Prime Pipeline. During this stage, several nlp techniques are applied to the question. This module tokenizes the question, identifies the pos tags, extracts Named Entities and types, identifies answer types and extracts question triples from the raw text data.

4.3.1. Tokenization, POS Tagging, Name Entity Recognition

We used Stanford CoreNLP in tokenization, POS tagging and named entity recognition tasks although there are several different libraries. We used SCNLP in this stage, because we used this library in the triple extraction stage, Other libraries may give us different outputs for tokenization, POS tagging, and NER. SCNLP may give us the wrong output for the dependency graph if we use the outputs of other libraries.

4.3.2. Question Classification and Answer Type Finding

Knowing the expected answer type is useful in question answering systems and finding the correct answer type is crucial. The correct answer type takes us to finding the correct answer to the question. A question answering system has to identify in which category the question fits. After that, system can eliminate any unnecessary answers based on the category.

Question classification plays an important role in QA systems. If the question classification part has an accurate definition of a question, the query itself will be accurate. Question classification has a significant role in the performance of the entire QA system. [31]

The simplest way to find the answer type is to assign an answer type based on the main question word. *When* means time or date, *Where* means location, *Who* means person. This approach works with simple factoid questions such as “Where is

Ankara?” However, real world questions are more complicated than this question. To find answer types to a question such as “What country in the world has the biggest area?” A QA system has to carry out more detailed processing. The answer type for this question is *country*, and *country* is a location.

There are several studies on question categories. The most recent and most well-known categories were proposed by Li and Roth [32], as shown in Table 6. Li and Roth proposed hierarchical answer types. These answer types have 6 coarse classes and 50 fine classes. Each coarse class contains a non-overlapping set of fine classes. Hermjakob [33] proposed 180 classes, which was the most comprehensive study. Gerber [34] categorized 18,000 questions with respect to their answer types. Hermjakob derived 115 QTargets (classes) from these categorizations as shown in Table 7. While analyzing questions such as “Who is the owner of CNN?” they added more QTargets with combinations of answer types, such as proper-person, proper-organization and ownership. Moldovan [35] proposed other hierarchical taxonomies shown in Table 8.

Table 6 Li and Roth’s Taxonomies (Question Classes)

Coarse Classes	Fine Classes
ABBREVIATION	abbreviation, expansion
DESCRIPTION	definition, description, manner, reason
ENTITY	animal, body, color, creation, currency, disease, event, food, instrument, language, letter, other, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word
HUMAN	description, group, individual, title
LOCATION	city, country, mountain, other, state
NUMERIC	code, count, date, distance, money, order, other, percent, period, speed, temperature, size, weight

Table 7 Some of the Qtargets (Question Classes) of Hermjakob

Energy	Quantity
Mass	Quantity
Proper	Person
Proper	Organization
Proper	Location
Plant	Flora
Sport	
Abbreviation	Expansion
At	Location
Proper	Island
Proper	Company
Substance	Liquid

Table 8 Moldovan Hierarchical Taxonomy

Question Class	Question	Answer Type
WHAT	Basic-what	Money/Number/
	Basic-who	Definition/Title
	Basic-when	/NNP/Undefined
	Basic-where	
WHO		Person
HOW	Basic-how	Manner
	How-many	Number
	How-long	Time/Distance
	How-much	Money/Price
	How-much	Undefined
	How-far	Distance
	How-tall	Number
	How-rich	Undefined
	How-large	Number
WHERE		Location
WHEN		Date
WHICH	Which-who	Person

	Which-where	Location
	Which-when	Date
	Which-what	NNP
Name	Name-who	Person
	Name-where	Location
	Name-what	Title/NNP
WHY		Reason
WHOM		Person

There are two different question classification approaches: Rule based and machine learning based. [36]

4.3.2.1. Rule Based Approach

The rule based approach needs predetermined rules. With these rules, the system attempts to find the category to which a question belongs. However, this approach has several advantages and disadvantages [37].

4.3.2.1.1. Advantages

- Rules can easily represent general knowledge about a problem domain in autonomous, relatively small chunks.
- Rules are a very natural knowledge representation method, with a high level of comprehensibility.
- Each rule is a discrete knowledge unit that can be inserted into, or removed from, the knowledge base regardless of any other technical detail.

4.3.2.1.2. Disadvantages

- Creating handwriting rules takes too much time.
- It is difficult to maintain a large rule base.
- The need to interview experts in order to create rules.
- If there is a missing value in the input data, the rules may not work.

- Inference efficiency is another problem. The performance of the inference engine may decrease.
- Previously learned classes cannot be exploited again.
- The general nature of rules may create problems in the interpretation of their scope while reasoning.
- Li and Roth provided an example, which shows the difficulty of rule-based approaches. All of the following samples are the same question that has been reformulated in different syntactical forms; therefore, it is difficult to apply a rule-based approach in some cases.
 - What tourist attractions are there in Reims?
 - What are the names of the tourist attractions in Reims?
 - What do most tourists visit in Reims?
 - What attracts tourists to Reims?
 - What is worth seeing in Reims?

Singhal et. al. [38] use these rules to find the question class;

- If a question starts with *Who* or *Whom*, the answer type is **Person**.
- If a question starts with *When*, the answer type is **Time, Date**.
- If a question starts with *Where*, the answer type is **Location**.
- If a question starts with *What* or *Which*, the head noun determines the class.
- If a question starts with *How-many*, *How-far*, *How-much*, the class will be **Quantity**.

4.3.2.2. Machine Learning Based Approach

Learning-based approaches perform classification by extracting some features from a text. The learning-based approach is more successful than the rule-based approach.

In the learning-based approach, the machine-learning model is designed and trained on an annotated corpus composed of labeled questions. Useful patterns are automatically captured from the corpus and added into taxonomies.

There are various classifiers to use in machine-learning based approaches, such as Neural Network, Naïve Bayes, Decision Tree, and Support vector machines.

The machine-learning based approach has several advantages over the rule-based approach. These advantages include:

- A short creation time
- No need for expert knowledge (automatic creation of classifier)
- Broader coverage; they can be obtain by providing new training examples.
- If required, the classifier can be flexibly reconstructed to fit to a new taxonomy.

4.3.3. Triple Extraction

Triples (sometimes called triplets) are basic data representations in the Semantic Web. Triples consist of a subject, predicate and object. Each triple contains a fact, so triples are called “Facts” in some data stores, such as YAGO.²⁰ Triples are structured in the form of a sentence. Each triple is extracted from a sentence; however, this does not mean every sentence contains only one triple.

Earlier, we stated that triples consist of a subject, predicate and object:

- Subject: that to which the triples refer
- Predicate: The relation between *Subject* and *Object*.
- Object: the property to which the *Subject* refers

Some examples of triples include:

- “Microsoft is located in Redmond.” = {Microsoft, located, Redmond}

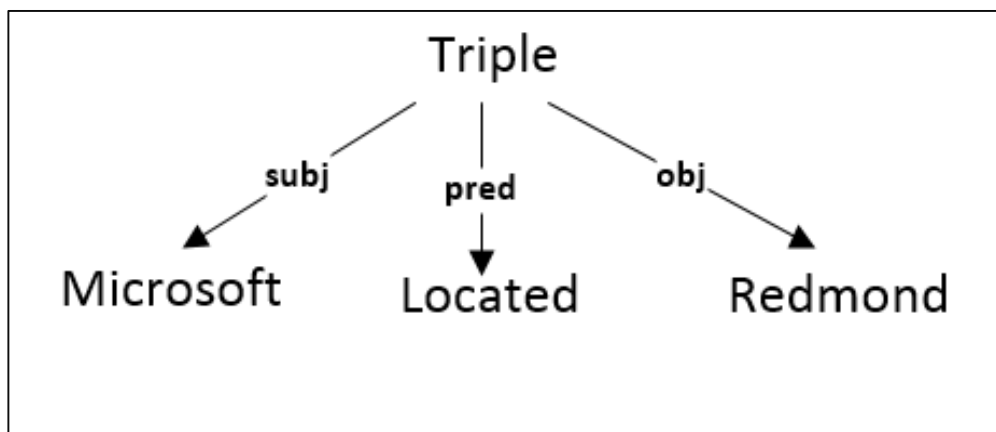


Figure 8 Sample Triple About Question Microsoft is location in Redmond.

²⁰ <http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/>

- “Google has 49,829 employees” = {Google, NumberOfEmployees, 49829}

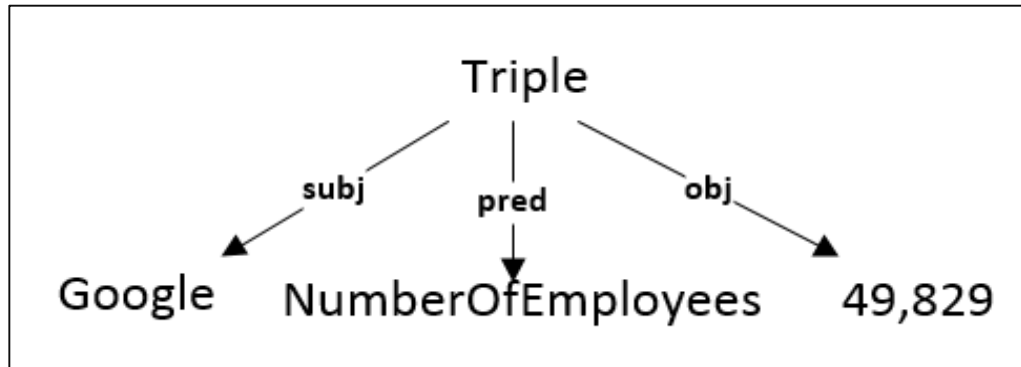


Figure 9 Sample Triple about Question Google has 49,829 employees

- “Albert Einstein was born in 14 March 1879” = {Albert Einstein, BirthDate, 14 March 1879}

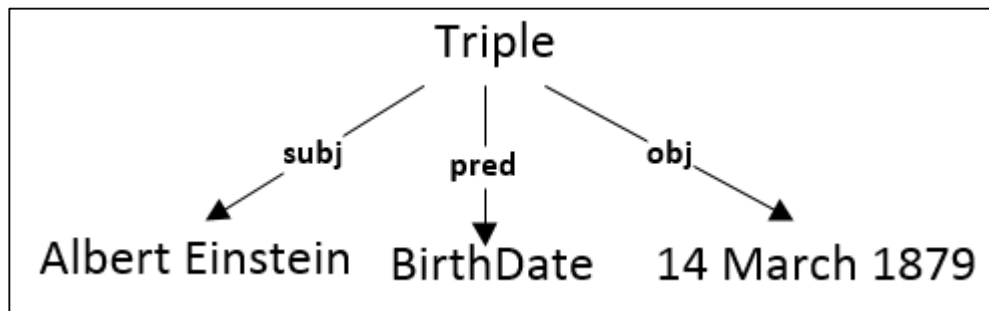


Figure 10 Sample Triple about Question Albert Einstein was born in 14 March 1879

To extract triples from a question, some natural language processing techniques are applied. We used Stanford CoreNLP²¹ library to extract triples from a question which was developed by the *Stanford Natural Language Processing Group*. SCNLP provides NLP tools including tokenization, POS tagging, lemmatization, named entity recognition, syntactic parsing and coreference resolution.

Triples can be extracted from a Dependency Graph and Parse tree.

²¹ <http://nlp.stanford.edu/software/corenlp.shtml>

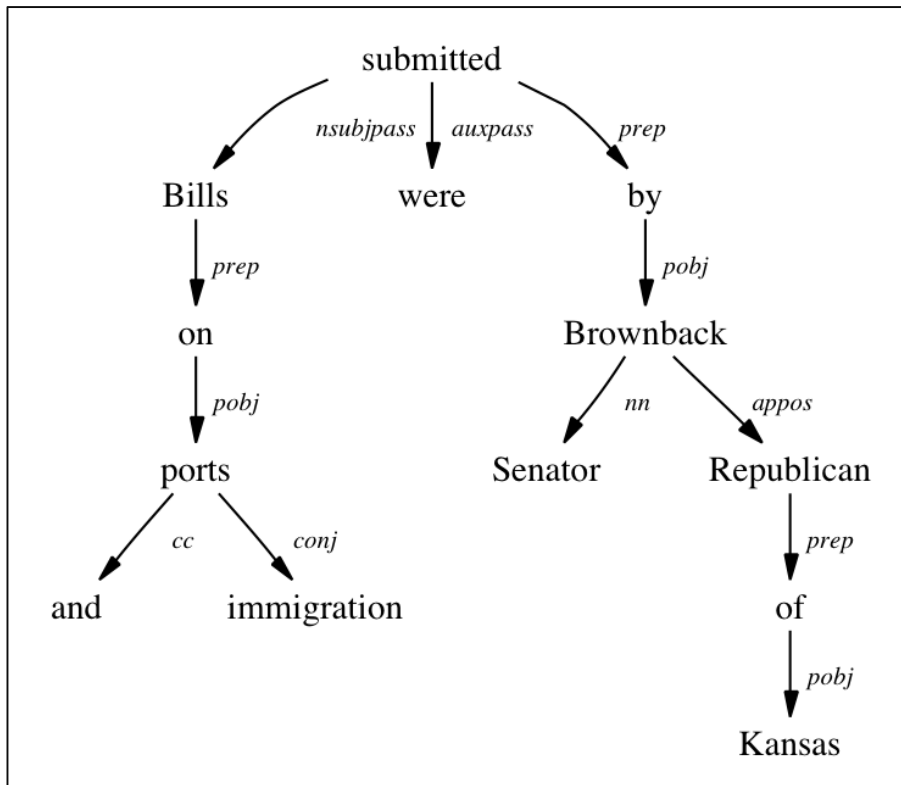


Figure 11 Sample Dependency Graph

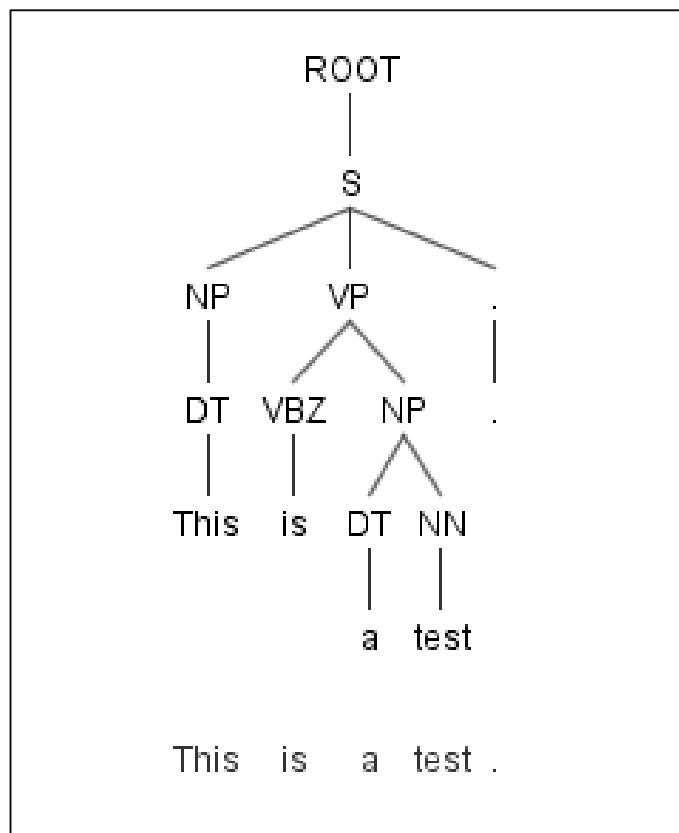


Figure 12 Sample Parse Tree

A parse tree splits phrases inside a sentence according to their grammatical relationship (Noun Phrase, Verb Phrase). A dependency graph works with relations of words inside a sentence. [39]

We used a dependency graph for extraction. We provide a sentence to Stanford CoreNLP and receive a dependency graph. The graph includes POS tags and relationships of words with parents and children. To extract triples, we traverse the tree starting from the root node to the leaf node. In each cycle, we examine the node and its children as a sub tree. If we can extract a triple from the tree, we add it to the list of triples.

A sample dependency graph extracted from Stanford CoreNLP is given below.

- “Which languages are spoken in Estonia?”

-> *spoken/VBN (root)*

-> *languages/NNS (nsubjpass)*

-> *Which/WDT (det)*

-> *are/VBP (auxpass)*

-> *Estonia/NNP (prep_in)*

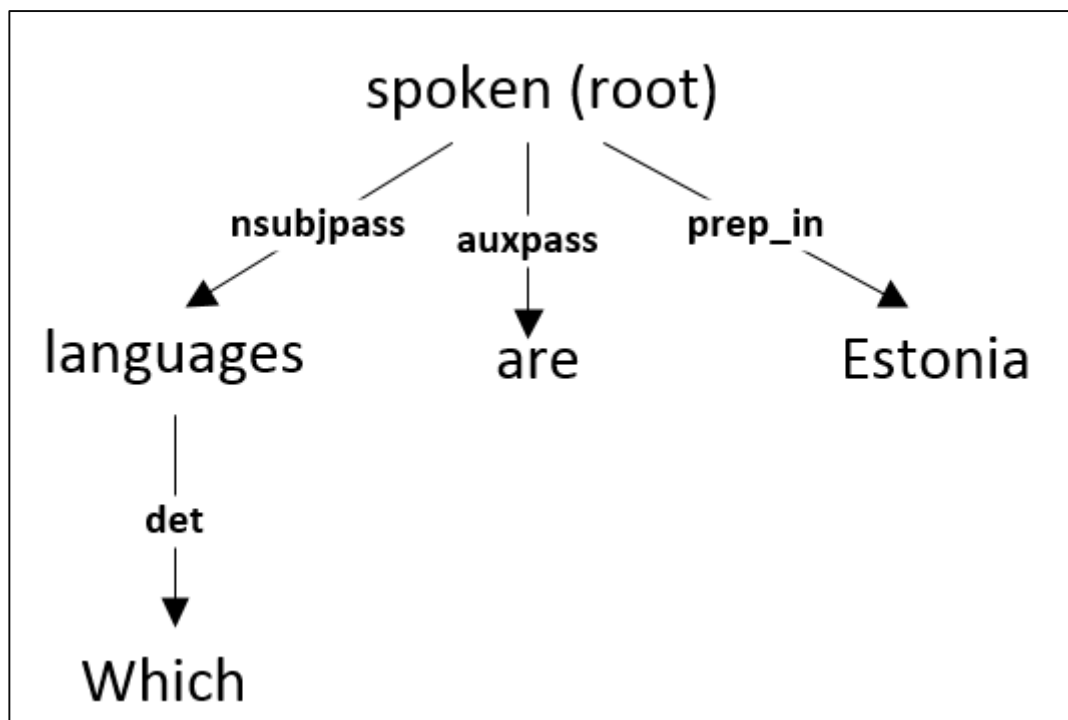


Figure 13 Dependency Graph of Question “Which languages are spoken in Estonia?”

To extract triples from a question, we use relations between words. We construct relation patterns to extract triples. Each pattern may produce a triple. To build a triple, we need a predicate, subject and object. There must be at least a subject or object. If we cannot find any of these, we do not add it to the triple list. While extracting the triple, we exclude some of the tags and relationships, namely *cop*, *det*, *aux*, *auxpass*, *expl*, *predet*, *preconj*, *mark*, *quantmod*, *parataxis*, *discourse*, and *case* relationships. These relationships are unnecessary for extracting triples; thus we ignore them. The entry point of the graph is *spoken*; generally, verbs are predicates of triples. We ignore the *det* and *auxpass* relations because of they have no effect on a sentence. While traversing the graph, we determine that *Estonia* is our subject because of the *prep_in* relation with the predicate. Our object is marked as *?x* and it is a variable to determine. In this question, only one triple is extracted.

From this graph, we extract this triple.

[Subject: Estonia/NNP] [Predicate: spoken/VBN] [Object: ?x]

Some questions contain more than one triple. For example, consider this question: “Who is the wife of the first president of the USA?” In this sentence, we can extract two different triples, which are:

- [Subject:USA] [Predicate:First President] [Object: ?x]
- [Subject: ?x] [Predicate: Wife] [Object ?y]

In this triple list, we have two different variables the first of which is $?x$ and second $?y$. Variable $?x$ is the first triple's object and also it is the second triple's subject. Variable $?x$ joins these two triples and finds the value of variable $?y$. The dependency tree works with formal sentences. We cannot extract triples from informal questions such as "Who is the Formula 1 race driver with the most races?"

4.4. Entity Mapping

After triple extraction, the next stage in our pipeline is entity mapping. In this stage, we are mapping the triple element (subject, predicate and object) onto the DBpedia ontology. For each triple element, we query DBpedia.

4.4.1. Finding Properties and Ontologies

For each element in the triple list, we searched in the DBpedia properties and ontologies. Before searching on DBpedia, we generate a synonym list for each predicate for each triple with WordNet.

Given predicate p , we use the following relations and scoring in WordNet. For example, predicate is language:

- For each synset in which p appears, we assign a score of 1.
 - Some of the synonyms of p
 - Sound
 - Spoken language
 - Speak
 - Talk
- For each hyponym in which p appears, we assign a score of $1 - \text{depth}/\text{maxDepth}$.
 - Some of the hyponyms of p
 - Read
 - Vocalize

- Chatter
- For each hypernym in which p appears, we assign a score of $0.1 - \text{depth}/\text{maxDepth}$ because hypernyms are a broad meaning of a word and communication is not a direct synonym of language.
 - Some of the hypernyms of p
 - Communication

After obtaining synonyms for each element from WordNet in predicates of the triples, we obtain property candidates from DBpedia. We first search for the original property in DBpedia; sometimes the original properties are directly matched with the DBpedia properties.

For example, in the sentence “What is the currency of Turkey?” the predicate is currency, and currency is in the DBpedia ontologies, and there is no need to query each synonym. *Currency* will be at the top of the list. We can map *currency* easily with the **dbo:currency** property and we can eliminate other elements in the list.

As mentioned earlier, we use the Lemon-model in Prime. WordNet synsets are mapped with DBpedia ontology in Lemon. For example, in the sentence “Where is Microsoft located?” SCNLP gives us “Microsoft” as a proper noun and located as the verb. If we write a sentence thus: “where is microsoft located” output of the SCNLP is will be different from our proper question. In this example, microsoft starts with small caps and there is no question mark. SCNLP gives us microsoft as a noun and located as an adjective. With these different outputs, we have to search WordNet with every POS tag “Verb” and “Adjective.”

Instead of searching in WordNet, first we search in the Lemon-Model.

For the word located, the Lemon-Model outputs 17 ontology classes and properties: *country, locationCity, locationCountry, range, city, location, locatedInArea, region, state, country, district, province, department, arrondissement, canton, closestCity*. After that, we add these to the mapping list.

For another question, “Who is the writer of The Lord of the Rings?” Our predicate in this question is **writer**, and if we search for writer in the Lemon-Model, it returns

only one ontology, which is the writer’s itself. Therefore, we cannot rely only on the Lemon-Model.

Neither WordNet nor Lemon-model is useful alone. Because of the best matching property and classes, we used both together.

4.4.2. Finding Name Entities

For each of the named entities and subjects in the triple list, we query DBpedia and find the matching entities called resources in DBpedia. For the question, “What is the currency of Turkey?” the named entity “Turkey” is extracted with SCNLP and it is also the subject of a triple. We search with the name and label the attribute in DBpedia. To query DBpedia, we generate a SPARQL query given in Figure 14.

```
SELECT DISTINCT ?subject ?name
WHERE {
  { ?subject foaf:name ?name . ?name bif:contains "Turkey" }
  UNION
  { ?subject rdfs:label ?name . ?name bif:contains "Turkey" } .
  { ?subject foaf:name ?name . FILTER(lang(?name) = "" || langMatches(lang(?name),
"en")) }
  UNION
  { ?subject rdfs:label ?name . FILTER(lang(?name) = "" || langMatches(lang(?name),
"en")) }
}
LIMIT 1000
```

Figure 14 SPARQL Query to find Name Entities

This query returns the top 1000 resources containing “Turkey” in its name and label properties. DBpedia returns results from all localized editions, such as English, German, French, Spanish, etc. Therefore, we add a filter for languages because we only want English DBpedia results. This filter obtains only the results from the English edition of DBpedia *langMatches(lang(?name), "en")*.

After querying DBpedia, we calculate a string similarity score for each result. While calculating the scores, we assign a score for the greatest common subsequence over the length of the word. [40] If the word score is 1.0f, the resource is an exact match with the named entity. If the score is less than 1.0f, then the resource is partially matched with the named entity.

Every matched resource is added to the mapping list. However, the mapped list can be quite large; therefore, we set a limit to the matched resource list of five resources. These resources will be used during the SPARQL generation stage to generate candidate queries.

4.5. SPARQL Generation

We cannot obtain answers from DBpedia with natural language. Therefore, we have to talk with its language. In order to query Triple Stores (RDF Store) such as DBpedia or MusicBrainz, we have to build a SPARQL query from the question asked by the user. Earlier in the question analysis stage, we extracted the question triples from the question. After this stage, we mapped the elements inside triples onto “DBpedia Ontology” and “Properties” and found the named entities.

In order to query DBpedia, we create candidate queries from the mapped properties and entities (resources). For each variable, we create a SPARQL query and add it to the candidate query list. Since we know the answer type of question, we find predicates with their types in DBpedia. For example, if our answer type is *number*, we only receive properties of type *xsd:int*, *xsd:double*, *xsd:float*, and *xsd:decimal* in the subject’s or object’s properties.

For a given question, “What is the currency of Turkey?” the generated valid query is given below.

```

SELECT DISTINCT ?currency (SAMPLE(?label) as ?label)
WHERE {
dbpedia:Turkey dbpedia-owl:currency ?currency .
OPTIONAL {
{ ?currency rdfs:label ?label . FILTER (lang(?label)="en" || lang(?label)="") }
UNION
{ ?currency foaf:name ?label. FILTER (lang(?label)="en" || lang(?label)="") } . }
GROUP BY ?currency
LIMIT 1000

```

Figure 15 SPARQL Query for Question “What is the currency of Turkey?”

A brief explanation of the query is:

- *SELECT DISTINCT ?currency (SAMPLE(?label) as ?label)* with this line we determine the result set. With the *Distinct* keyword, we eliminate duplicate values for *?currency*. For *?currency*, here we have to specify *?currency*, *?label* in the *WHERE* and *OPTIONAL* clause. The *SAMPLE* keyword is used for returning arbitrary values from the result set.
- *WHERE* is the query triple pattern from which we want to find the result. A *WHERE* clause can be constructed in a multiple triple pattern or only in one triple pattern. Each triple pattern has to contain a *Subject*, *Predicate* and an *Object*. In order to write a multiple triple pattern in a *WHERE* clause, we have to put “.” (dot) after each triple pattern.
- *OPTIONAL* is, as it can be understood from the name, the place where we put optional query patterns. Sometimes the value we want to query cannot be available for all RDF triples. For a given query, we put a filter if *name* and *label* of the triple contains the English language. If we cannot find the English values for *rdfs:label* and *foaf:name*, our query still returns a result set for a given query, but the query returns only the value of *?currency*. If we do not put these triple patterns inside the *OPTIONAL* part to return the result, all the triple patterns must be matched with RDF triples.

- The *GROUP BY* keyword is used when we want to retrieve aggregate values for a result set. In the query above, we group the result set with *?currency* and with the *SAMPLE* keyword, we return an arbitrary value for the result set.
- The *LIMIT* keyword restricts the number of returned rows in an executed query. In the query above, we restricted the maximum returned result to 1000.

4.6. Query Execution

After creating the candidate queries, we start to query DBpedia. We iterate over the list until we get result from DBpedia. If we do receive a result from a query we exit the loop and return result to user. We set the limit maximum 100 of candidate queries. If we cannot find any answer for every query, we return an error to the user.

The result of a query can be either *Resource* or *Literal*. The *Resource* result points to another DBpedia resource. For example, the result of the given question “What is the currency of Turkey?” is http://dbpedia.org/resource/Turkish_Lira, and it is pointing to another result in DBpedia. If we navigate to http://dbpedia.org/resource/Turkish_Lira in a browser, it redirects to http://dbpedia.org/page/Turkish_Lira.

Literal results are simple plain values. For example, the result of the given question “When did Michael Jackson die?” is “2009-06-25”. This result is a literal type of date. A list of literal types is given in Table 9:

Table 9 List of Literal Types in SPARQL

Datetime	String	Decimal	Double	Float
Integer	Boolean	Long	Short	Byte

If the result of a question is *Literal*, we simply return the value. If the result of the question is *Resource*, we receive an abstract text, name or image of the result. To obtain additional information from the result, we prepare the following query:

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>

SELECT * WHERE { { OPTIONAL{?url dbpedia-owl:abstract ?abstract}
OPTIONAL{?url dbpedia-owl:thumbnail ?thumbnail}
OPTIONAL{?url rdfs:label ?name}
OPTIONAL{?url foaf:isPrimaryTopicOf ?wikiUrl}
FILTER(?url=<http://dbpedia.org/resource/Turkish_lira>) }
FILTER(lang(?abstract)="en"&&lang(?name)="en") }
```

Figure 16 SPARQL Query for Getting Additional Information of a DBpedia Resource

CHAPTER 5

EVALUATION

5.1. Testing Prime over QALD-5 Questions

In order to evaluate Prime, we used QALD-5 [41] (Question Answering over Linked Data-5) challenge questions over DBpedia. The challenge contains two different question sets called training and test. The training question set contains 340 questions and the test question set contains 59 questions which are given in Appendix A.

The question sets contain questions in natural language in seven languages and manually created keywords for each language. In the training set, each question has a SPARQL query that corresponds to a question and the answers to the question. After the challenge has finished, the answers and SPARQL queries of the test set are published.

The QALD-5 challenge contains different question types, including factoid, list, yes/no and hybrid question types. Our system supports only simple factoid questions, so we excluded some question types, namely:

- List questions
 - “Give me all cosmonauts.”
 - “Give me a list of all bandleaders that play trumpet.”
- Yes/No questions
 - “Does Breaking Bad have more episodes than Game of Thrones?”
 - “Is proinsulin a protein?”
- Questions contains YAGO [42] classes
 - “Which U.S. state has been admitted latest?”
 - “Sean Parnell is the governor of which U.S. state?”
- Some of the questions are marked as “Out of Scope” and have no answer, so these questions are excluded.

- “What is the most beautiful painting?”
- “In which studio did the Beatles record their first album?”
- Hybrid Questions
 - There are 40 hybrid questions. This question type is QALD specific. To answer these questions, the system has to analyze structured (RDF) and unstructured text data.
 - “Who was vice-president under the president who authorized atomic weapons against Japan during World War II?”
- Complex Questions
 - “Which Chess players died in the same place they were born in?”
 - “Which other weapons did the designer of the Uzi develop?”
- Questions containing abbreviations
 - “Where was JFK assassinated?”
 - “Which U.S. state has the abbreviation MN?”
- Questions containing comparison operators such as “more than,” “bigger than,” “most,” etc. These comparison operators have to be implemented to the SPARQL query in order to receive correct results.
 - “Which German cities have more than 250000 inhabitants?”
 - “What is the longest river?”
- Questions containing highest/lowest values.
 - “What was Brazil’s lowest rank in the FIFA World Ranking?”
 - “Which of Tim Burton's films had the highest budget?”
- Some questions contain adjectives that must be considered while generating SPARQL queries.
 - For example, in the question “What is the second highest mountain on Earth?” we have to add ORDER BY DESC for the elevation property of mountain resources. Moreover, FILTER 1 and LIMIT 1 keywords must be added to the query to receive only one result.
- In some questions, we have to count the results and obtain the items.
 - For example, in the question “Which countries have more than ten caves?” we have to add GROUP BY for countries and add HAVING(COUNT(?x) > 10) to the query to receive results.

- Some questions ask a specific date.
 - For example, in the question “Which presidents were born in 1945?” we have to match 1945 to the date and implement a comparison to the literal values.

After excluding any necessary questions, the remaining questions in the train list number 116 and the remaining questions in the test list number 24. The evaluation results are based on these questions.

5.2. Test Results of the System

Test results are calculated with QALD-5 Challenge Evaluation measures. For each question q , we calculated precision, recall, and F-1 measure as follows:

$$Recall(q) = \frac{\text{Number of Correct System Answers}}{\text{Number of Gold Standart Answers}}$$

$$Precision(q) = \frac{\text{Number of Correct System Answers}}{\text{Number of System Answers}}$$

$$F - 1 \text{ Measure}(q) = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

For each question, precision, recall and F-1 Measure values were computed. Overall Precision, Recall and the F-1 measure values were computed as the average mean of the Precision, Recall and the F-1 measure values for all questions.

We computed Precision, Recall and the F-1 Measure for Prime in Table 10. The results in Table 11 are obtained from the QALD-5 challenge final report.

Column Description of Evaluation Results

- Processed states for how many of the questions the system provided a query and/or answer.
- Right specifies how many of these questions were answered with an F-1 measure of 1.
- Partial specifies how many of the questions were answered with an F-1 measure strictly between 0 and 1.
- Recall, Precision and F-1 report the measures with respect to the number of processed questions.

Table 10 Prime’s Evaluation Results of QALD-5 Test And Train Question Set

Question Set	Processed	Right	Partial	Recall	Precision	F-1
Training	116	46	5	0.41	0.42	0.41
Test	24	8	2	0.36	0.40	0.37

Table 11 QALD-5 Evaluation Results

Name	Processed	Right	Partial	Recall	Precision	F-1
Xser[39]	42	26	7	0.72	0.74	0.73
APEQ[46]	26	8	5	0.48	0.40	0.44
QAnser[42]	37	9	4	0.35	0.46	0.40
SemGraphQA[40]	31	7	3	0.32	0.31	0.31
YodaQA[41]	33	8	2	0.25	0.28	0.26

As shown in Table 11, 5 participants attended the QALD-5 multilingual question answering over the DBpedia challenge. [43]

The best scoring system Xser [44] answered more than half of the questions. Xser obtained answers from the knowledge base in two steps. First, they analyzed the question and extracted the predicate-argument structure from the question. This allowed them to generate queries that were knowledge base independent. After that, they map the instantiate of the query with the knowledge base. Although their method requires training the data, they received the best score in the challenge.

APEQ [43] ranked second in the challenge. Although the system scored far behind Xser and processed fewer questions than the others, its recall and precision was good. It uses the graph traversal technique. They first analyze the question in terms of phrase structure and then they extract the main entity, followed by exploring the RDF graph using this entity to discover relationships with other entities mentioned inside the question. They measure the path of the graph and the best scoring entity is returned.

QAnswer [45] did not use any training data and only attempted to answer those questions marked as “onlydbo,” which means only use DBpedia ontology in challenge questions. The main problems for QAnswer include mapping wrong entities, not being able to compare dates, failing to interpret questions and mapping properties to types.

SemGraphQA [46] first matches words inside a question with DBpedia entities, properties and classes, followed by building a syntactic graph of the question with the Stanford Parser. The syntactic graph is converted to syntactic-semantic graphs using relations between words and mapped elements to the knowledge base. They assume each word inside the graph is either an entity or a relation and they connect to an implicit entity. For each syntactic-semantic graph, they create all possible semantic graphs. They scored semantic graphs and built SPARQL queries. Their method works and does not need training data and can be ported to other knowledge bases besides DBpedia. In the challenge, they processed 31 questions and only answered 7 questions correctly and 2 questions partially correct. There were some errors while answering questions such as the entity is not found in 7 questions, relation is not found in 25 questions, and type is not found in 4 questions. As we can see the errors, their system is not correctly finding relations between words and entities because the real meaning of a relation does not correspond to its label in the knowledge base.

YodaQA [47] first generates a bag-of-features such as keywords, phrases, etc. followed by a candidate answer by searching the knowledge base according to these features. Finally, they applied a scoring algorithm to find the answer according to the

features. Their system processed 33 questions and answered 10 questions; however, their precision and recall were lower than any other system. This is because their method make searches based on bag-of-features and score the results based on clues such as keywords, key phrases, and concept clues crisply matching Wikipedia titles and lexical answer types. Moreover, YodaQA is optimized to work with unstructured data such as Wikipedia. [48]

As shown in Table 10, we evaluated Prime for both test and training questions. In the training question set, Prime correctly answered 46 questions and 5 questions were partially answered with 42% precision. In the test questions, Prime correctly answered 8 questions and 2 questions were partially answered with 40% precision. The test set results of Prime were lower than the training set results because in the training set, Prime answered nearly half of the questions correctly. Moreover, there are multiple similar questions inside the training questions such as “How many employees does Google have?” and “How many employees does IBM have?” Prime produced the correct answers to those questions. Because of this, our precision and recall was higher than the test questions.

When we compare evaluation results with other participants in QALD-5, Prime processed only 24 of the test questions. Four participant systems processed more than 30 questions. However, given the correct answers, Precision and Recall are nearly the same as the other participants. Some questions were answered by Prime, but most of the questions were not answered. Moreover, Prime did not supply a valid SPARQL query for each question; thus, the system could not query the data source and produce an answer.

The questions could not be answered by Prime due to the following reasons:

- Question classification error.
 - The system could not identify the answer type of questions correctly. For the given question, “Which of Tim Burton’s films had the highest budget?” the expected answer type is *movie*, but the system returns *organization*. When the wrong expected answer type is identified, the system maps variables inside triples to the wrong entity type.

- Matching wrong entities and properties in the data source and ambiguity problems in questions
 - For the question “When was the Titanic completed?” the question refers to the ship RMS Titanic (not the movie) and our system did not find correct entity for the ship RMS Titanic.
- Extracted triple patterns do not match with the data source.
 - For the question “To which countries does the Himalayan mountain system extend?” our system produced a triple to [Subject: Himalayan Mountain System] [Predicate: Extend] [Object: Country]. However, the triple should be mapped to [Subject: Himalayan] [Predicate: Country] [Object: ?x].

5.3. Performance of the System

The performance of the system is directly related to DBpedia and a number of generated SPARQL queries. Some of the queries took too long to be completed in DBpedia’s SPARQL endpoint because we are sometimes querying free text to find entities and properties. DBpedia returns property query results quickly. If we are searching common names using free text such as “Michael,” the query takes too long to complete and the system raises an exception.

Prime generates candidate queries to answer a question. Sometimes the number of these queries may exceed hundreds. Although we limit the maximum candidate query size to 100, the correct query can be at the end of the list. While we are querying the candidate queries, it takes too much time to complete, so this delay is the response time.

Some performance metrics are shown in Table 12;

Table 12 Prime's Performance Metrics

Question	Generated SPARQL Queries	Answer Found in (*)nth Query	Answer Found in * Seconds
Who created Goofy?	146	1	7
What is the time zone of Salt Lake City?	24	1	5
Who founded Intel?	97	1	7
What is the official color of the University of Oxford?	22	1	6
Which actor played Chewbacca?	38	1	6
In which city does the Chile Route 68 end?	70	1	6

As we can see from these metrics, Prime generates multiple queries to even a simple question since for each variable in triples, Prime finds synonyms of predicates and finds named entities to expand the candidate query list. Generally, in simple questions, an answer was found in the first query. This is due to the entities and predicates being directly matched with DBpedia entities and ontologies. Prime can answer questions in a minimum of 5 seconds and questions are answered between 5 and 10 seconds by Prime. The response speed is directly related to DBpedia and Internet speed. Some queries take too long to complete in DBpedia, especially free text search queries while finding named entities. If a query takes too long to complete and is still working, Prime stops the execution and attempts to execute the next query; however, this operation increases the response time.

CHAPTER 6

CONCLUSION

Question answering systems have received much attention in the last few years and these systems are the upper level of traditional search engines. Finding the correct answer from traditional search engine results is not unlike the proverbial “finding a needle in a haystack” due to the size of the data growing day by day and the doubling every two years of the data we create. The necessity of quickly accessing data has been the reason for accelerated research in this area.

A number of search engines, such as Google, have already implemented question answering in their search results in some languages; however, these systems are not only search engines which work like a web site. Apple Siri²², has question answering capabilities, in addition to Google Now²³ and Microsoft Cortana.²⁴ These search engines are only in our phones. With Amazon Echo,²⁵ these systems are found in our homes.

Question answering systems not only search for textual data from the Web and show them to users. These systems have capabilities such as an Intelligent Assistant and can make mathematical calculations, show sports results, acquire information on weather conditions, show stock prices and even predicting the future. Question answering systems are the future of search engines and Intelligent Assistants.

²² <http://www.apple.com/ios/siri/>

²³ <https://www.google.com/landing/now/>

²⁴ <http://windows.microsoft.com/en-us/windows-10/getstarted-what-is-cortana>

²⁵ <http://www.amazon.com/dp/B00X4WHP5E>

To achieve this, researchers conduct a great amount of research on natural language processing, information retrieval and information extraction. Without this research, these applications would be dreams.

In this thesis, we explained (1) what a question answering system is, (2) how they worked and (3) developed, and tested a question answering system called Prime. Prime has the capability of answering simple factoid questions. It analyzes a question asked in natural language and obtains answers from DBpedia. To query DBpedia, Prime extracts triples from a question and generates SPARQL queries. Variables inside extracted triples must be mapped to DBpedia ontology to obtain the correct results. DBpedia ontology is fixed in size but the properties inside a resource (DBpedia entity or page) are not fixed. These variables vary from resource to resource. A resource type of person and a resource type of location have different properties. Moreover, sometimes the same type of resource has different properties. It is difficult to find the desired property due to the property in the resource possibly not being the same as the desired answer. Human language is ambiguous and very many words have multiple meanings. FREyA [28] implemented feedback and clarification dialogs to overcome ambiguity. A user can ask a question with a different syntactic structure. In a question in which a user wants to learn the height of a person, a question such as “How tall is Michael Jordan?” may be asked. The resource of Michael Jordan does not contain a property such as *tall*, but it contains the *height* property. We have to map *tall* to *height* in order to find the correct answer to the question. We used predefined ontologies and mappings to overcome this difficulty in addition to implementing manually created lexicons [27] to improve performance. Even using predefined ontologies and mappings were insufficient.

Prime’s test results against the QALD-5 challenge are promising, but our correct answer count was low as we processed only 24 questions in the test set. The best scoring system Xser [44] answered more than half of the questions as they use a technique different from other systems to the query knowledge base. Our system showed better precision and recall than YodaQA [47] and SemGraphQA [46] since we processed only 24 questions, correctly answered eight questions and partially answered two questions. Implementing other question types, such as List, Definition,

and Complex questions, missing features such as comparison operators, extracting highest/lowest values and detecting abbreviations, etc. can increase the number of correct answers.

The main conclusion of this thesis is that building a question answering system is a challenging task. Although unanswered questions outnumber correctly answered questions, the results are promising. Prime's capability can be extended taking into account the challenges in unanswered questions with new implementations.

FUTURE WORK

The current architecture of the system can answer simple factoid questions but some missing features should be implemented to answer a wider range of questions.

The triple extraction method should be improved to extract triples from complex questions. Improving the triple extraction method affects the performance gain and increases the answer accuracy rate system-wide.

Our system supports only Factoid questions; it cannot answer List, Definition or Complex questions. Support for other questions types other than Factoids can be added to expand the question count. In addition, we excluded some questions as they contained comparison operators, highest/lowest values, YAGO classes etc. Adding support to these features in the system can answer more questions.

Although the English edition of DBpedia is the largest DBpedia edition, some information can be found inside other editions of DBpedia. Languages other than English can be implemented in the system to search in other editions of DBpedia, such as the Spanish, German and French language versions. Adding multiple language support expands the coverage of the system.

REFERENCES

1. **Bringing big data to the enterprise.** 08-13-2015. <http://www-01.ibm.com/software/data/bigdata/what-is-big-data.html>
2. **Jr., B. F. G., Wolf, A. K., Chomsky, C., & Laughery, K. (1961).** “*Baseball: an automatic question-answerer.*” AFIPS Joint Computer Conferences, 219–224.
3. **Daniel Jurafsky and James H. Martin (2008).** “*Speech and Language Processing (2nd Edition) (Prentice Hall Series in Artificial Intelligence).*” Prentice Hall, 2nd Edition
4. **Berners-Lee, T., Hendler, J., & Lassila, O. (2001).** “*The Semantic Web.*” Scientific American, 284(5), 34–43.
5. **Grigoris Antoniou, & Frank van Harmelen. (2009).** “*A Semantic Web Primer.*” The Knowledge Engineering Review, 24(04), 415.
6. **Gruber, T. (1995).** “*What is an Ontology?*” Knowledge Acquisition.
7. **Noy, N., & McGuinness, D. (2001).** “*Ontology development 101: A guide to creating your first ontology.*” Development, 32, 1–25.
8. **Linked Data Project.** 08-04-2015. <http://linkeddata.org/>.
9. **The Linking Open Data Cloud Diagram.** 08-13-2015. <http://lod-cloud.net/>
10. **Berners-Lee, T. (2007).** “*Design Issues -- Linked Data.*”
11. **Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mende, P. N., Hellmann S., Morsey M., Kleef P., Auer S., Bizer, C. (2012).** “*DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia.*” Semantic Web, 1, 1–5.
12. **Logogram.** 18-13-2015. <https://en.wikipedia.org/wiki/Logogram>

13. **Grishman, R., & Sundheim, B. (1996).** *“Message Understanding Conference-6: A Brief History.”* Proceedings of the 16th Conference on Computational Linguistics, 1, 466–471.

14. **Miller, G. a. (1995).** *“WordNet: a lexical database for English.”* Communications of the ACM, 38(11), 39–41.

15. **McCrae, J., Spohr, D., & Cimiano, P. (2011).** *“Linking lexical resources and ontologies on the semantic web with lemon.”* Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6643 LNCS(PART 1), 245–259.

16. **Colby, K. M., Weber, S., & Hilf, F. D. (1971).** *“Artificial Paranoia.”* Artificial Intelligence, 2(1), 1–25.

17. **Woods, W. a., Kaplan, R. M., & Nash-Webber, B. L. (1972).** *“The Lunar Sciences Natural Language Information System: Final Report.”* Technical Report BBN Report, Bold Beranek and Newman Inc., Cambridge, Massachusetts, 2378.

18. **Woods, W. a. (1973).** *“Progress in natural language understanding: an application to lunar geology.”* Proceedings of the National Computer Conference and Exposition on AFIPS '73, 441–450.

19. **Simmons, R. F. (1965).** *“Answering English questions by computer: a survey.”* Communications of the ACM, 8(1), 53–70.

20. **Lehnert, Wendy G. (1978).** *“The process of question answering: A computer simulation of cognition.”* Lawrence Erlbaum Associates

21. **Lehnert, W. G. (1977).** *“A conceptual theory of question answering.”* Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 1, 158–164.

22. **Scha, R. J. H. (1977).** *“Philips question-answering system PHLIQAI.”* ACM SIGART Bulletin, (61), 26.

23. **Katz, B. (1988).** *“Using English for Indexing and Retrieving.”* Proceedings of the 1st RIAO Conference on User-Oriented Content-Based Text and Image Handling (RIAO '88).

24. **Katz, B. (1997).** *“From Sentence Processing to Information Access on the World Wide Web.”* AAAI Spring Symposium on Natural Language Processing for the World Wide Web, 77–86.

25. **Ferrucci, D. a. (2012).** *“Introduction to This is Watson.”* IBM Journal of Research and Development, 56(3.4), 1:1–1:15.

26. **Lopez, V., Pasin, M., & Motta, E. (2005).** *“AquaLog: an ontology-portable question answering system for the semantic web.”*

27. **Unger, C., & Cimiano, P. (2011).** *“Pythia: Compositional meaning construction for ontology-based question answering on the semantic web.”* Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6716 LNCS, 153–160.

28. **Damljanovic, D., Agatonovic, M., & Cunningham, H. (2012).** *“FREyA: An interactive way of querying linked data using natural language.”* Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7117 LNCS, 125–138.

29. **Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014).** *“The Stanford CoreNLP Natural Language Processing Toolkit.”* Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 55–60.7

30. **Schlaefler, N. (2005).** *“Pattern Learning and Knowledge Annotation for Question Answering.”*

31. **Pasca, M., & Harabagiu, S. M. (2001).** *“High Performance Question / Answering.”* Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval Pages, 366 – 374.

32. **Li, X., & Roth, D. (2006).** *“Learning question classifiers: the role of semantic information.”* Natural Language Engineering, 12(03), 229.

33. **Hermjakob, U. (2001).** *“Parsing and question classification for question answering.”* In Proceedings of the workshop on ARABIC language processing status and prospects - (Vol. 12, pp. 1–6). Morristown, NJ, USA: Association for Computational Linguistics.

34. **Laurie Gerber. (2001).** *“A QA Typology for Webclopedia.”*

35. **Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Goodrum, R., Girju, R., & Rus, V. (1999).** *“Lasso: A tool for surfing the answer net.”* Trec.

36. **Skowron, M., & Araki, K. (2004).** “*Evaluation of the new feature types for question classification with support vector machines.*” IEEE International Symposium on Communications and Information Technology, 2004. ISICIT 2004., 2.
37. **Prentzas, J., & Hatzilygeroudis, I. (2007).** “*Categorizing approaches combining rule-based and case-based reasoning.*” Expert Systems, 24(2), 97–122.
38. **Singhal, A., S. Abney, M. Bacchiani, M. Collins, D. Hindle, and F. Pereira. (2000).** “*AT&T at TREC-8.*” In E. Voorhees, editor, Proceedings of the 8th Text Retrieval Conference, NIST.
39. **Marneffe, M. De, & Manning, C. D. (2010).** “*Stanford typed dependencies manual.*” 20090110 Stanford, 40(September), 1–22. Retrieved from http://nlp.stanford.edu/downloads/dependencies_manual.pdf
40. **Hakimov, S., Hakan, T., Marlen, A., & Erdogan, D. (2013).** “*Semantic Question Answering System over Linked Data using Relational Patterns.*”
41. **The Fifth Open Challenge on Question Answering Over Linked Data. (2015).** Retrieved from <http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/5/documents/qald-5.pdf>
42. **Suchanek, F. M., Kasneci, G., & Weikum, G. (2007).** “*Yago - A Core of Semantic Knowledge.*” Proceedings of the 16th International Conference on World Wide Web - WWW '07, 697.
43. **Unger, C., Forascu, C., Lopez, V., Ngomo, A.-C. N., Cabrio, E., Cimiano, P., & Walter, S. (2015).** “*Question Answering over Linked Data (QALD-5).*”
44. **Kun Xu, Feng, Y., & Dongyan Zhao. (2014).** “*Xser@QALD-4: Answering Natural Language Questions via Phrasal Semantic Parsing.*”
45. **Ruseti, S., Alexandru Mirea, Traian Rebedea, Stefan, & Trausan-Matu. (2015).** “*QAnswer - Enhanced Entity Matching for Question Answering over Linked Data.*” CLEF.
46. **Romain Beaumont, Grau, B., & Ligozat, A.-L. (2015).** “*SemGraphQA@QALD-5: LIMSI participation at QALD-5@CLEF.*”
47. **Baudis, P.** “*YodaQA: A Modular Question Answering System Pipeline.*”

48. **Baudiš, P., & Šedivý, J.** “*Modeling of the Question Answering Task in the YodaQA System.*”

APPENDICES A

QALD-5 Training Questions

	Question
1.	Give me all cosmonauts.
2.	In which country does the Ganges start?
3.	When is the movie Worst Case Scenario going to be in cinemas in the Netherlands?
4.	Which German cities have more than 250000 inhabitants?
5.	Who was John F. Kennedy's vice president?
6.	Who is the mayor of Berlin?
7.	How many students does the Free University in Amsterdam have?
8.	What is the second highest mountain on Earth?
9.	Give me all professional skateboarders from Sweden.
10.	When was Alberta admitted as province?
11.	To which countries does the Himalayan mountain system extend?
12.	Give me a list of all bandleaders that play trumpet.
13.	Which countries have more than ten caves?
14.	What is the total amount of men and women serving in the FDNY?
15.	Who produces Orangina?
16.	Who is the Formula 1 race driver with the most races?
17.	Give me all world heritage sites designated within the past two years.
18.	Who is the youngest player in the Premier League?
19.	Give me all members of Prodigy.
20.	What is the longest river?
21.	Does Breaking Bad have more episodes than Game of Thrones?
22.	Give me all cars that are produced in Germany.
23.	What were the main discoveries of the Mars rover Curiosity?
24.	Give me all people that were born in Vienna and died in Berlin.
25.	Is proinsulin a protein?
26.	How tall is Michael Jordan?
27.	What is the capital of Canada?
28.	Who is the governor of Wyoming?
29.	Do Prince Harry and Prince William have the same parents?
30.	Who was the father of Queen Elizabeth II?
31.	Which U.S. state has been admitted latest?
32.	How many languages are spoken in Turkmenistan?
33.	Sean Parnell is the governor of which U.S. state?
34.	Give me all movies directed by Francis Ford Coppola.
35.	Give me all actors starring in movies directed by William Shatner.
36.	Are tree frogs a type of amphibian?

37.	What is the birth name of Angela Merkel?
38.	Give me all Methodist politicians.
39.	How often did Jane Fonda marry?
40.	Give me all Australian nonprofit organizations.
41.	In which military conflicts did Lawrence of Arabia participate?
42.	Who developed Minecraft?
43.	What is the melting point of copper?
44.	Give me all sister cities of Brno.
45.	How many inhabitants does Maribor have?
46.	Give me all companies in Munich.
47.	How tall is Claudia Schiffer?
48.	List all games by GMT.
49.	Who founded Intel?
50.	Who is the husband of Amanda Palmer?
51.	Give me all breeds of the German Shepherd dog.
52.	Which cities does the Weser flow through?
53.	Which countries are connected by the Rhine?
54.	Which professional surfers were born in Australia?
55.	What is the average temperature on Hawaii?
56.	In which UK city are the headquarters of the MI6?
57.	Which other weapons did the designer of the Uzi develop?
58.	Who created Goofy?
59.	Was the Cuban Missile Crisis earlier than the Bay of Pigs Invasion?
60.	Give me all islands that belong to Japan.
61.	Who invented the zipper?
62.	What is the ruling party in Lisbon?
63.	What are the nicknames of San Francisco?
64.	Which Greek goddesses dwelt on Mount Olympus?
65.	When were the Hells Angels founded?
66.	Give me all Apollo 14 astronauts.
67.	What is the time zone of Salt Lake City?
68.	Which U.S. states are in the same time zone as Utah?
69.	Give me the capitals of all countries in Africa.
70.	Give me a list of all lakes in Denmark.
71.	How many missions does the Soyuz programme have?
72.	Did Socrates influence Aristotle?
73.	Give me all Danish movies.
74.	Give me all launch pads operated by NASA.
75.	Which instruments does Cat Stevens play?
76.	Which ships were called after Benjamin Franklin?
77.	Who are the parents of the wife of Juan Carlos I?
78.	How many employees does Google have?
79.	Did Tesla win a nobel prize in physics?
80.	Give me all cities in New Jersey with more than 100000 inhabitants.
81.	Is Rita Wilson the wife of Tom Hanks?
82.	When was the Statue of Liberty built?
83.	In which U.S. state is Fort Knox located?

84.	How many children did Benjamin Franklin have?
85.	When did Michael Jackson die?
86.	Which Chess players died in the same place they were born in?
87.	List the children of Margaret Thatcher.
88.	Who was called Frank The Tank?
89.	Was Margaret Thatcher a chemist?
90.	Was Marc Chagall a jew?
91.	Which museum exhibits The Scream by Munch?
92.	Give me all books by William Goldman with more than 300 pages.
93.	Which books by Kerouac were published by Viking Press?
94.	Give me a list of all American inventions.
95.	How high is the Mount Everest?
96.	Who created the comic Captain America?
97.	How many people live in the capital of Australia?
98.	What is the largest city in Australia?
99.	Who composed the music for Harold and Maude?
100.	Which films starring Clint Eastwood did he direct himself?
101.	In which city was the former Dutch queen Juliana buried?
102.	Is Egypt's largest city also its capital?
103.	Where is the residence of the prime minister of Spain?
104.	Which U.S. state has the abbreviation MN?
105.	Show me all songs from Bruce Springsteen released between 1980 and 1990.
106.	Which movies did Kurosawa direct?
107.	What is the founding year of the brewery that produces Pilsner Urquell?
108.	Who wrote the lyrics for the Polish national anthem?
109.	Give me all B-sides of the Ramones.
110.	Who painted The Storm on the Sea of Galilee?
111.	Which country does the creator of Miffy come from?
112.	For which label did Elvis record his first album?
113.	Give me the birthdays of all actors of the television show Charmed.
114.	How many employees does IBM have?
115.	Which states border Illinois?
116.	In which country is the Limerick Lake?
117.	Which television shows were created by John Cleese?
118.	Which mountain is the highest after the Annapurna?
119.	In which films directed by Garry Marshall was Julia Roberts starring?
120.	Which bridges are of the same type as the Manhattan Bridge?
121.	Was U.S. president Jackson involved in a war?
122.	Give me all communist countries.
123.	Which awards did Douglas Hofstadter win?
124.	Who is the daughter of Robert Kennedy married to?
125.	Which U.S. state has the highest population density?
126.	What is the currency of the Czech Republic?
127.	Which countries adopted the Euro?
128.	What is the area code of Berlin?
129.	Which countries have more than two official languages?

130.	Who is the owner of Rolls-Royce?
131.	Through which countries does the Yenisei river flow?
132.	When did Latvia join the EU?
133.	Which monarchs were married to a German?
134.	When was the Battle of Gettysburg?
135.	Which river does the Brooklyn Bridge cross?
136.	What is the highest mountain in Australia?
137.	Give me all soccer clubs in Spain.
138.	What is the official language of Suriname?
139.	Who is the mayor of Tel Aviv?
140.	Who designed the Brooklyn Bridge?
141.	Which telecommunications organizations are located in Belgium?
142.	Is Frank Herbert still alive?
143.	What is the highest place of the Urals?
144.	Who is the editor of Forbes?
145.	Give me all companies in the advertising industry.
146.	How many countries are there in Europe?
147.	What did Bruce Carver die from?
148.	Give me all libraries established earlier than 1400.
149.	Which presidents were born in 1945?
150.	Give me all federal chancellors of Germany.
151.	Who was the wife of U.S. president Lincoln?
152.	Who developed the video game World of Warcraft?
153.	What is the official website of Tom Cruise?
154.	List all episodes of the first season of the HBO television series The Sopranos.
155.	Who produced the most films?
156.	Give me all people with first name Jimmy.
157.	In which city did John F. Kennedy die?
158.	Is there a video game called Battle Chess?
159.	Which mountains are higher than the Nanga Parbat?
160.	Who created Wikipedia?
161.	Give me all actors starring in Last Action Hero.
162.	Which software has been developed by organizations founded in California?
163.	Which companies work in the aerospace industry as well as in medicine?
164.	Is Christian Bale starring in Velvet Goldmine?
165.	Give me the websites of companies with more than 500000 employees.
166.	Which actors were born in Germany?
167.	Which caves have more than 3 entrances?
168.	Was the wife of president Lincoln called Mary?
169.	Give me all films produced by Hal Roach.
170.	Give me all video games published by Mean Hamster Software.
171.	Which languages are spoken in Estonia?
172.	How many Aldi stores are there?
173.	Which capitals in Europe were host cities of the summer olympic games?
174.	Who was the 16th president of the United States?
175.	How many films did Hal Roach produce?

176.	Which music albums contain the song Last Christmas?
177.	Give me all books written by Danielle Steel.
178.	Which airports are located in California, USA?
179.	Which states of Germany are governed by the Social Democratic Party?
180.	Give me all Canadian Grunge record labels.
181.	Which country has the most official languages?
182.	In which programming language is GIMP written?
183.	Who produced films starring Natalie Portman?
184.	Give me all movies with Tom Cruise.
185.	In which films did Julia Roberts as well as Richard Gere play?
186.	Give me all female given names.
187.	Who wrote the book The Pillars of the Earth?
188.	How many films did Leonardo DiCaprio star in?
189.	Give me all soccer clubs in the Premier League.
190.	In which U.S. state is Mount McKinley located?
191.	When was Capcom founded?
192.	Which organizations were founded in 1930?
193.	What is the highest mountain?
194.	Was Natalie Portman born in the United States?
195.	Which budget did the first movie of Zdenek Sverak have?
196.	How many big fires struck Paris during the Middle Ages?
197.	Is Pamela Anderson a vegan?
198.	How often was Michael Jordan divorced?
199.	What is the most beautiful painting?
200.	Give me all animal species that live in the Amazon rainforest.
201.	How many inhabitants does the largest city in Canada have?
202.	In which studio did the Beatles record their first album?
203.	Who was the first to climb Mount Everest?
204.	How many programming languages are there?
205.	What is the official color of the University of Oxford?
206.	How many gold medals did Michael Phelps win at the 2008 Olympics?
207.	To which artistic movement did the painter of The Three Dancers belong?
208.	Give me all animals that are extinct.
209.	Does Abraham Lincoln's death place have a website?
210.	How deep is Lake Placid?
211.	Give me the grandchildren of Bruce Lee.
212.	Who is the youngest Darts player?
213.	Where was Bach born?
214.	In which countries can you pay using the West African CFA franc?
215.	What are the top-10 action role-playing video games according to IGN?
216.	What is the most frequent cause of death?
217.	Does the Isar flow into a lake?
218.	Give me all films produced by Steven Spielberg with a budget of at least \$80 million.
219.	Give me all writers that won the Nobel Prize in literature.
220.	Give me all taikonauts.
221.	How many pages does War and Peace have?

222.	What is the bridge with the longest span?
223.	Give me all actors called Baldwin.
224.	Who is the tallest player of the Atlanta Falcons?
225.	Which rivers flow into a German lake?
226.	How many James Bond movies are there?
227.	Which rockets were launched from Baikonur?
228.	Which pope succeeded John Paul II?
229.	Give me all Dutch parties.
230.	When is Halloween?
231.	Give me all Swedish oceanographers.
232.	Give me all actors who were born in Berlin.
233.	What was the last movie with Alec Guinness?
234.	Which poet wrote the most books?
235.	How many languages are spoken in Colombia?
236.	What does IYCM stand for?
237.	What was Brazil's lowest rank in the FIFA World Ranking?
238.	Give me the capitals of all countries that the Himalayas run through.
239.	Which actor played Chewbacca?
240.	Which ingredients do I need for carrot cake?
241.	Is Cola a beverage?
242.	Who has Tom Cruise been married to?
243.	Which of Tim Burton's films had the highest budget?
244.	How heavy is Jupiter's lightest moon?
245.	Which actor was casted in the most movies?
246.	Is James Bond married?
247.	Give me all Australian metalcore bands.
248.	Give me all actors who were born in Paris after 1950.
249.	When was Carlo Giuliani shot?
250.	Who are the four youngest MVP basketball players?
251.	Which companies have more than 1 million employees?
252.	Give all swimmers that were born in Moscow.
253.	Who was called Rodzilla?
254.	Show me the book that Muhammad Ali wrote.
255.	How many museums does Paris have?
256.	Which city has the most inhabitants?
257.	Which city has the least inhabitants?
258.	Give me all the TV shows with Neil Patrick Harris.
259.	Who wrote The Hunger Games?
260.	Show a list of soccer clubs that play in the Bundesliga.
261.	What country is Mount Everest in?
262.	Who is the founder of Penguin Books?
263.	Which programming languages influenced Javascript?
264.	Did Che Guevara have children?
265.	List all the musicals with music by Elton John.
266.	Show me all the breweries in Australia.
267.	When was the Titanic completed?
268.	How much did Pulp Fiction cost?

269.	How many airlines are there?
270.	Who played Agent Smith in Matrix?
271.	How much carbs does peanut butter have?
272.	Which book has the most pages?
273.	Which bridges cross the Seine?
274.	Who is the mayor of the capital of French Polynesia?
275.	When did Dracula's creator die?
276.	What is the location of the Houses of Parliament?
277.	Show me all English Gothic buildings in Kent.
278.	Who was the pope that founded the Vatican Television?
279.	What airlines are part of the SkyTeam alliance?
280.	What is the total population of Melbourne, Florida?
281.	Which airports does Air China serve?
282.	In which year was Rachel Stevens born?
283.	Where was JFK assassinated?
284.	How many politicians graduated from Columbia University?
285.	Which scientist is known for the Manhattan Project and the Nobel Peace Prize?
286.	What is the highest volcano in Africa?
287.	Which beer originated in Ireland?
288.	What are the specialities of the UNC Health Care?
289.	Who is the owner of Facebook?
290.	From which region is the Melon de Bourgogne?
291.	Who was influenced by Socrates?
292.	Who was president of Pakistan in 1978?
293.	Give me English actors starring in Lovesick.
294.	Give me all types of eating disorders.
295.	Who was married to president Chirac?
296.	What is the largest metropolitan area in Washington state?
297.	Where in France is sparkling wine produced?
298.	Where did Hillel Slovak die?
299.	What is the timezone in San Pedro de Atacama?
300.	In which city does the Chile Route 68 end?
301.	Who has vice-president under the president who authorized atomic weapons against Japan during World War II?
302.	In which town was the assassin of Martin Luther King born?
303.	Which anti-apartheid activist was born in Mvezo?
304.	How many Golden Globe awards did the daughter of Henry Fonda win?
305.	Which recipients of the Victoria Cross died in the Battle of Arnhem?
306.	Where did the first man in space die?
307.	How old was Steve Jobs' sister when she first met him?
308.	Which members of the Wu-Tang Clan took their stage name from a movie?
309.	Which writers had influenced the philosopher that refused a Nobel Prize?
310.	Under which king did the British prime minister that signed the Munich agreement serve?
311.	Who composed the music for the film that depicts the early life of Jane Austin?

312.	Who succeeded the pope that reigned only 33 days?
313.	On which island did the national poet of Greece die?
314.	Which horses did The Long Fellow ride?
315.	Of the people that died of radiation in Los Alamos, whose death was an accident?
316.	Which building owned by the Bank of America was featured in the TV series MegaStructures?
317.	Which buildings in art deco style did Shreve, Lamb and Harmon design?
318.	Which birds are protected under the National Parks and Wildlife Act?
319.	Which country did the first known photographer of snowflakes come from?
320.	List all the battles commanded by the lover of Cleopatra.
321.	Are the Rosetta Stone and the Gayer-Andersen cat exhibited in the same museum?
322.	Which actress starring in the TV series Friends owns the production company Coquette Productions?
323.	Gaborone is the capital of which country member of the African Union?
324.	When was the composer of the opera Madame Butterfly born?
325.	Which street basketball player was diagnosed with Sarcoidosis?
326.	For which movie did the daughter of Francis Ford Coppola receive an Oscar?
327.	Which city does the first person to climb all 14 eight-thousanders come from?
328.	At which college did the only American actor that received the César Award study?
329.	Did Napoleon's first wife die in France?
330.	How old is James Bond in the latest Bond book by William Boyd?
331.	What eating disorder is characterized by an appetite for substances such as clay and sand?
332.	What is the native city of Hollywood's highest-paid actress?
333.	In which city does the former main presenter of the Xposé girls live?
334.	Who plays Phileas Fogg in the adaptation of Around the World in 80 Days directed by Buzz Kulik?
335.	Who is the front man of the band that wrote Coffee & TV?
336.	Which Chinese-speaking country is a former Portuguese colony?
337.	What is the largest city in the county in which Faulkner spent most of his life?
338.	In which year did the Hungarian-American actor called "The King of Horror" make his first film?
339.	Under which pseudonym did Charles Dickens publish some of his books?
340.	A landmark of which city is the home of the Mona Lisa?

QALD-5 Test Questions

	Question
1.	Give me all ESA astronauts.
2.	Give me all Swedish holidays.
3.	Who is the youngest Pulitzer Prize winner?
4.	Which animals are critically endangered?
5.	Which soccer players were born on Malta?
6.	Did Arnold Schwarzenegger attend a university?
7.	Which programming languages were influenced by Perl?
8.	Is Barack Obama a democrat?
9.	How many children does Eddie Murphy have?
10.	Who is the oldest child of Meryl Streep?
11.	Who killed John Lennon?
12.	Which frequent flyer program has the most airlines?
13.	In which city is Air China headquartered?
14.	Which artists were born on the same date as Rachel Stevens?
15.	How many scientists graduated from an Ivy League university?
16.	Which types of grapes grow in Oregon?
17.	Who is starring in Spanish movies produced by Benicio del Toro?
18.	Who is the manager of Real Madrid?
19.	Give me the currency of China.
20.	Which movies starring Brad Pitt were directed by Guy Ritchie?
21.	How many companies were founded by the founder of Facebook?
22.	How many companies were founded in the same year as Google?
23.	Which subsidiary of Lufthansa serves both Dortmund and Berlin Tegel?
24.	How many airlines are members of the Star Alliance?
25.	Give me all spacecrafts that flew to Mars.
26.	Which musician wrote the most books?
27.	Show me everyone who was born on Halloween.
28.	Give me all Swiss non-profit organizations.
29.	In which country is Mecca located?
30.	What is the net income of Apple?
31.	What does the abbreviation FIFA stand for?
32.	When did the Ming dynasty dissolve?
33.	Which museum in New York has the most visitors?
34.	Is Lake Baikal bigger than the Great Bear Lake?
35.	Desserts from which country contain fish?
36.	What is the highest mountain in Italy?
37.	Where did the architect of the Eiffel Tower study?
38.	Which Greek parties are pro-European?
39.	What is the height difference between Mount Everest and K2?
40.	Who is the mayor of Rotterdam?
41.	In which city were the parents of Che Guevara born?
42.	How high is the Yokohama Marine Tower?
43.	Are Taiko a kind of Japanese musical instruments?

44.	How many ethnic groups live in Slovenia?
45.	List the seven kings of Rome.
46.	Who were the parents of Queen Victoria?
47.	Who is the heaviest player of the Chicago Bulls?
48.	Which volcanos in Japan erupted since 2000?
49.	Who is the tallest basketball player?
50.	Where was the "Father of Singapore" born?
51.	Which Secretary of State was significantly involved in the United States' dominance of the Caribbean?
52.	Who is the architect of the tallest building in Japan?
53.	What is the name of the Viennese newspaper founded by the creator of the croissant?
54.	In which city where Charlie Chaplin's half brothers born?
55.	Which German mathematicians were members of the von Braun rocket group?
56.	Which writers converted to Islam?
57.	Are there man-made lakes in Australia that are deeper than 100 meters?
58.	Which movie by the Coen brothers stars John Turturro in the role of a New York City playwright?
59.	Which of the volcanoes that erupted in 1550 is still active?

APPENDICES B
CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: ÖZKAN, Fatih

Date and Place of Birth: 02 October 1986, Ankara

Phone: +90 536 527 33 94

Email: fatihozk@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
B.Sc.	Anadolu University, Economics	2012
B.Sc.	Ahmet Yesevi University, Computer Engineering	2012
High School	Keçiören Kanuni Lisesi	2004

WORK EXPERIENCE

Year	Place	Enrollment
2014- ...	Seneka Yazılım	Senior Software Developer
2011-2014	Kale Yazılım	Team Leader
2009-2011	Türksat A.Ş.	Software Developer
2008-2009	MBA Danışmanlık	Software Developer
2006-2008	Simetri Yazılım	Software Developer

FOREIN LANGUAGES

English - Advanced