



JIMMA UNIVERSITY
JIMMA INSTITUTE OF TECHNOLOGY
FACULTY OF COMPUTING
GRADUATE PROGRAM IN INFORMATION TECHNOLOGY

**QUERY ANSWERING OVER THE WEB OF DATA: THE CASE
OF AFAAN OROMO**

By
ALEMISA ENDEBU ERGOSA

A THESIS SUBMITTED TO JIMMA INSTITUTE OF TECHNOLOGY, FACULTY OF
COMPUTING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

Jimma, Ethiopia

June, 2019

JIMMA UNIVERSITY
JIMMA INSTITUTE OF TECHNOLOGY
FACULTY OF COMPUTING
GRADUATE PROGRAM IN INFORMATION TECHNOLOGY

**QUERY ANSWERING OVER THE WEB OF DATA: THE CASE
OF AFAAN OROMO**

By

ALEMISA ENDEBU ERGOSA

Advisor Name: Melkamu Beyene (PhD)

Co-Advisor Name: Mr. Admas Abtew (MSc)

A THESIS SUBMITTED TO JIMMA INSTITUTE OF TECHNOLOGY, FACULTY OF
COMPUTING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

Jimma, Ethiopia

June, 2019


DECLARATION

I, ALEMISA ENDEBU ERGOSA hereby declare that the work presented here in is original work done by me and has not been published or submitted elsewhere for the requirement of a degree in any other Universities. Any literature or work done by other and cited within this thesis has given due acknowledgement and listed in the reference section.

A THESIS SUBMITTED BY

ALEMISA ENDEBU ERGOSA	_____	<u>June, 2019</u>
	SIGNATURE	DATE

APPROVED BY ADVISORS:

ADVISOR: Melkamu Beyene (PhD)		<u>June, 2019</u>
	SIGNATURE	DATE

CO-ADVISOR: <u>Mr.Admas Abtew (MSc)</u>	_____	<u>June, 2019</u>
	SIGNATURE	DATE

Name and Signature of the Board Examiners for Approval

<u>NAME</u>	<u>SIGNATURE</u>
1. Teklu Urgessa (PhD)	_____
2. Getachew Mamo (PhD)	_____
3. Tesfu Mokenen (MSc)	_____

DEDICATION:

I dedicated this thesis to my beloved Parents and Family for their endless love, support, encouragement and sacrifices.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my heartfelt gratitude to all those who helped me to make my thesis work a success. First and foremost, I would like to praise and thank the supernatural power and creature of the entire universe almighty God for helping me to realize this work.

My heartfelt thanks go to my thesis advisor **Melkamu Beyene (PhD)** for his uninterrupted follow up from the title selection to the completion of this thesis work. Really, his friendly treatment and fruitful suggestions were admirable and marvelous. Thank you my mentor for your endless enthusiasm throughout this work. You have devoted your priceless time and endeavor effort on me to accomplish this work on time. I would also like to express my sincere thanks to my Co-Advisor **Mr. Admas Abtew (MSc)** for his encouragement, inspiring advice and moral support in carrying out this thesis work.

My special thanks go to my friend **Fikedu Wayessa (Fike)**, I have no words to express what you did for me and God bless you and your life endlessly. I would like to thank all my friends, thank you for your understanding and inspiration in many, many moments of crisis. Your friendship makes my life a wonderful experience. I cannot list all the names of my friends here, but you are always on my mind. In addition, I would like to thank all staff members of Jimma University, Faculty of Computing, for your kind help during my stay and I feel happy to thank Jimma institute of technology for sponsoring me.

There are no words to express my gratitude and thanks to My Beloved parents and Family Members. Their love has been the major unworldly support in my life.

As a closing word, I would like to thank each and every individual who have been a source of support and encouragement and helped me to achieve my goal and complete my thesis work successfully.

Thank you, Lord for always being there for me!

Table of Contents

DECLARATION	i
DEDICATION:.....	ii
ACKNOWLEDGEMENTS	iii
LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF ACRONYMS AND ABBREVIATIONS.....	ix
ABSTRACT.....	x
CHAPTER ONE	1
1. INTRODUCTION	1
1.1. Background	1
1.2. Statement of the Problem.....	4
1.3. Research Questions.....	7
1.4. Objectives of the Study	7
1.4.1. General Objective	7
1.4.2. Specific Objective	7
1.5. Scope of the Study	7
1.6. Significance of the Study	10
1.7. Thesis Organization	11
CHAPTER TWO	12
2. LITERATURE REVIEW	12
2.1. Overview.....	12
2.2. Web Search engine	12
2.2.1. Web crawling.....	12
2.2.2. Indexing Documents	13
2.2.3. Search for documents.....	13
2.3. Semantic web Based Approaches	16
2.3.1. The structure of the Semantic Web.....	17
2.3.2. The Semantic web application area.....	20
2.3.3. Challenges for Semantic web Search Engine.....	20
2.4. Ontology Based Approaches.....	22
2.4.1. Knowledge Representation (KR)	22

2.4.2.	Ontology to represent knowledge	22
2.5.	Basic components of ontology	23
2.6.	Building Ontology search	24
2.8.	Related Work	27
2.8.1.	Summary	31
CHAPTER THREE	32
3.	Overview of Afaan Oromo	32
3.1.	Afaan Oromo Writing System	32
3.1.1.	Afaan Oromo alphabets	32
3.1.2.	Afaan Oromo Punctuation Marks	33
3.1.3.	Afaan Oromo Abbreviations	33
3.2.	Word Categories of Afaan Oromo	34
3.3.	Afaan Oromo Morphology.....	37
3.4.	Questions in Afaan Oromo	38
CHAPTER FOUR	39
4.	MODEL QUERY ANSWERING OVER WEB OF DATA FOR AFAAN OROMO.....	39
4.1.	Proposed System Architecture Overview	39
4.1.1.	Web of data source identification	42
4.1.2.	Converting into a Single Coherent Knowledge Graph.....	43
4.1.3.	Knowledge graph	43
4.1.4.	Querying the Web of Data	44
4.1.5.	Query Translation	45
4.1.6.	User Query Interface.....	48
CHAPTER FIVE	56
5.	EXPERIMENTATION AND EVALUATION	56
5.1.	Overview	56
5.2.	Experiment procedure	56
5.3.	Prototype	66
5.4.	Evaluation parameters.....	69
5.5.	Discussion of the result	71
CHAPTER SIX	74
6.	CONCLUSION AND FUTURE WORK.....	74

6.1. Overview	74
6.2. Conclusion	74
6.3. Future works	75
References.....	76

LIST OF TABLES

Table 2-1: Summary of related works	31
Table 3-1: Personal Pronouns of Afaan Oromo.....	37
Table 4-1: Parts of speech (POS) representation	47
Table 5-1: Sample CSV test data files	57
Table 5-2: Classes, Object Properties and Data properties in the domain.....	60
Table 5-3: Ontology classes.....	60
Table 5-4: Object properties in Education ontology.....	63
Table 5-5: Data type properties in Education ontology	63
Table 5-6: Sample query tested for experimentation.....	65
Table 5-7: The number of query and their evaluation parameters.....	70

LIST OF FIGURES

Figure 1.1: Classical Search Engine	2
Figure 1.2 : Query Answering System.....	2
Figure 1.3: Query Answering System for Afaan Oromo	4
Figure 1.4: Design Science Research Framework	8
Figure 2.1: An illustration of PageRank	16
Figure 2.2: The layered structure of the Semantic web	17
Figure 4.1: Proposed System Architecture	41
Figure 5.1: Class Hierarchy using Protégé for education domain.	62
Figure 5.2: Afaan Oromo Query answering over web of data prototype	66
Figure 5.3: Afaan Oromo Query Answering over web of data display the result	67
Figure 5.4: User interface for checking Validation	68
Figure 5.5: Recall, Precision and F-Measure graph.....	72

LIST OF ACRONYMS AND ABBREVIATIONS

CSV	Comma Separated Value
DSR:	Design Science Research
IR:	Information Retrieval
IT:	Information Technology
KB:	Knowledge base
QA:	Query Answering
OWL:	Ontology Web Language
RDF:	Resource Description Framework
SE:	Search Engine
SPARQL:	Simple protocol and RDF Query Language
TF:	Term Frequency
TF-IDF:	Term Frequency- inverse Document Frequency
URL:	Uniform Resource locator
WWW:	World Wide Web

ABSTRACT

Nowadays, with the development of the Semantic Web, a lot of new structured data has become available on the Web in the form of knowledge bases. Query Answering over Web of Data is a field that has been widely explored in research area. Most current QA systems query on Web of Data, in one language (namely English). The existing approaches are not designed to be easily adaptable to new knowledge bases and languages. One of them is Afaan Oromo language. Research in Afaan Oromo Query Answering is still limited and has not reached the same level of English Query Answering due to the Afaan Oromo language specific challenges. Most of existing research in Afaan Oromo Query Answering has not explored the field of Query Answering on the web of data, and has mainly focused on natural language processing (NLP) and information retrieval from unstructured Afaan Oromo documents.

The web is developing rapidly towards the notation of linked data where the data is linked by exploiting the semantic web technologies and standards. However, the variety of linked-data sources and their high heterogeneity make it difficult for humans to search and discover relevant information. As linked data is in RDF format, the standard approach would be to run structured queries in triple-pattern based languages like SPARQL, but only expert programmers are able to precisely specify their information needs. Users who have no knowledge with semantic web cannot express their queries in SPARQL. This problem can be fixed by using natural language interfaces that translate natural language queries to SPARQL.

This study aims to make a step towards supporting Afaan Oromo Query Answering over Web of Data. The approach we propose to translate Afaan Oromo Natural language queries, to SPARQL. We have tested by using sample ontology on education domain to translate the user query to RDF triple and retrieve an answer from a RDF knowledge base. The proposed approach can process only on simple sentence query. The experimentation shows that the performance is on the average 66.03 % Recall, 76.17% Precision, and 71.63% F-measure.

Keywords: *Query Answering, Ontology, RDF triple, OWL, Web of Data, and SPARQL.*

CHAPTER ONE

1. INTRODUCTION

1.1. Background

Currently, the Web has become the main source of information where lots of terabytes of data are added every day in all fields. The amount of information in the web has grown largely[1]. The number of users in the internet has recently reached more than 4 billion though the world and, in average, each of them sends more than one email per day without considering other issues[2][3].

The World Wide Web (WWW) was created by Tim Berners-Lee with the vision of connecting people[5]. It did not take long for WWW to become a global phenomenon and become the backbone for communication on the global level. So much information has been uploaded on the web that, information could be found just about anything on the web[6][7]. Nowadays, with the emergence of the web of data (i.e. rapid development of data published using semantic web technologies on the web), query answering systems over Web of Data have received wide attention[8]. Query answering allow users to express arbitrarily complex information needs in an easy way and intuitive fashion[6].

Query answering process of fetching specie facts, answer questions, give advice, or compose reports that satisfy users is the goal of next generation search engines¹. Figure 1.1 and 1.2 shows the difference between classical search engines and query answering systems for the user query *when is Jimma University is established*

¹Towards the Next Generation Information Retrieval

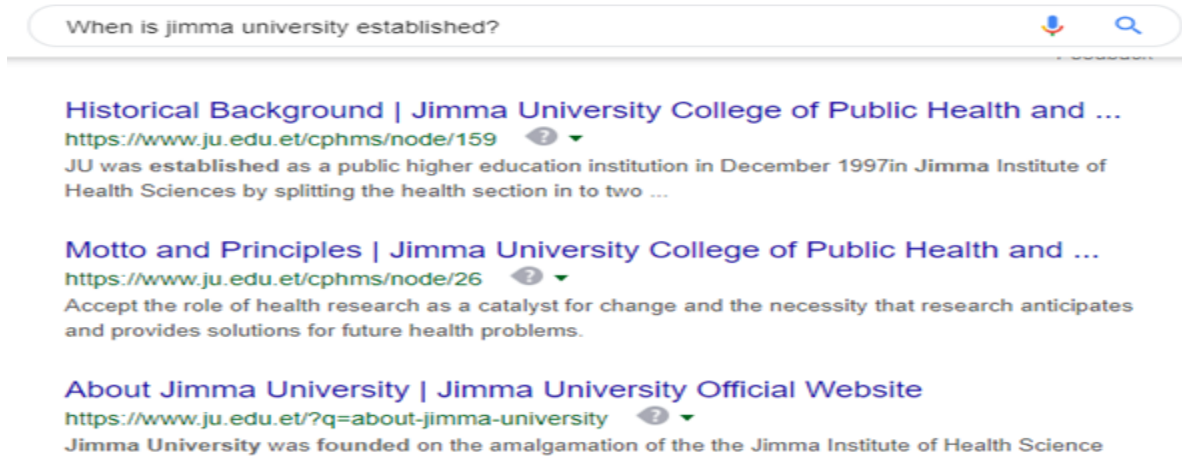


Figure 1.1: Classical Search Engine

As shown in figure 1.1, classical search engines (i.e. keyword based searches) retrieve a list of webpages or documents that match with keywords in the user query. But users are expected to extract specific information from this webpages. However, query answering systems understand the meaning of the user query and give answers in response to a user query.

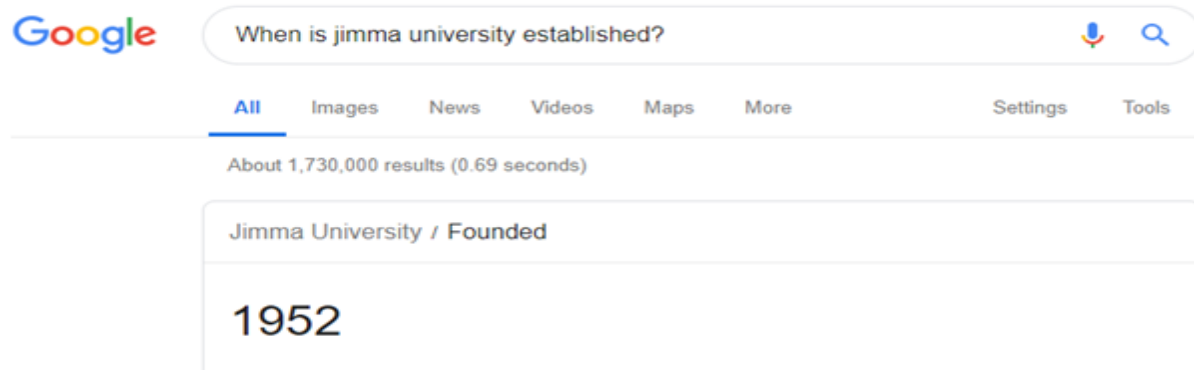


Figure 1.2 : Query Answering System

In web of data, concepts in documents are linked to related concepts in other documents by using a new standard, the Resource Description Framework (RDF) and Ontology Web Language (OWL). On the other way, in the traditional web terms in documents are linked by a set of

keywords. Thus, the essential idea behind web of data is the collection of concepts that are linked together not the collection of documents. Searching on web of data for semantic query are more efficient than other web engines because in this era of busy life everyone wants an exact answer to his/her query which only semantic web can provide[9].

Query answering systems like Google use the web of data (structured data distributed in the web) as a source answer queries. For instance, Google use the Google Knowledge Graph², one of the structured data source to give answers for such queries. The goal of query answering as presented in[4], is to allow users to ask query in natural language, using their own terminology and receive a terse answer. For query answer over web of data, user asks query in natural language. The process starts by linguistically analyzing. The next step is to classify the query according to defined question category. Then, the SPARQL query is generated. An ontology resource can be used for matching items generated in the process. Finally, when the SPARQL query is generated, the interrogation of the linked data is done and generates the exact answer of the user query

Furthermore, with this rapid development of the web of data, there are many RDF datasets published as linked data [6]. There are also datasets published in other formats like xml, csv etc[10]. However, there are still gaps that need further research. For instance users including computer experts, have difficulties to write queries using query language like SPARQL to access this data. On the other hand, due to the diversity and high heterogeneity of the data, it is difficult for humans to select relevant resources and discover useful information. Query answering over Web of Data is aimed at eliminating those “gaps”, which attempts to allow the users to access those structured data with by providing natural language queries. To fulfill this aim Query Answering Over web of data has become a hot research topic [11].

2. <https://developers.google.com/knowledge-graph/>

There have been many attempts to design a query answering systems in English. However, as far as the researcher's knowledge is concerned, no research is conducted to design a query answering mechanism Over the Web of Data for Afaan Oromo Language despite researches conducted to design classical search for Afaan Oromo documents [12][13] , That means the area query answering over web of data has not been studied for Afaan Oromo language. Therefore, we cannot get answers for the query in figure-1.3

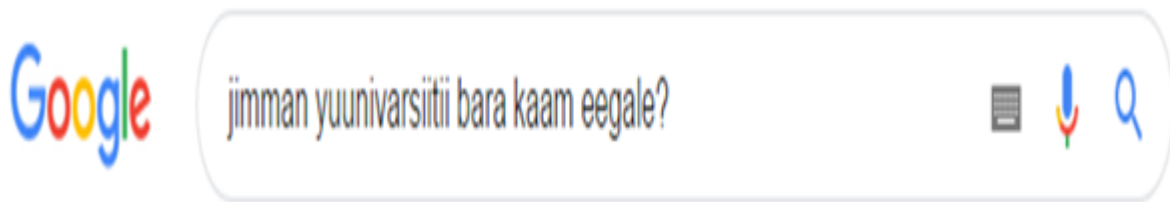


Figure 1.3: Query Answering System for Afaan Oromo

In this study, we proposed a Query Answering system over Web of Data for Afaan Oromo. The aim is answering user query from web of data source for natural language questions given in Afaan Oromo. There are many language specific problems when designing such systems. One of the main problems lies in translating the user's information needs into a form such that they can be evaluated using standard semantic web query processing.

1.2. Statement of the Problem

Afaan Oromo is one of the major African languages that is widely spoken and used in most of Ethiopian and parts of other neighbor countries like Kenya, Somalia and Sudan [14]. It is used by Oromo people, who are the largest ethnic group in Ethiopia which amounts more than 35% of the total population. Currently, various works are published in journals on area of Natural Language Processing (NLP) and Information Retrieval(IR)[15][16][17] related to the Afaan Oromo language. However, with amount of data available on the web increasing rapidly in

Afaan Oromo document, our information needs cannot be met by even today's state of the art of web technologies.

Tesfaye Guta, proposed Afaan Oromo search engines [12] , but they depend on keyword-based search which do not represent the meaning of the content of the documents and any relationship between them. The serious issue of keyword-based search (also called Syntactic search) engines such as Google, Gmail, Bing, and Yahoo is the loss of keyword semantics, which gives many irrelevant results for the users[18].

Traditional keyword based search engines suffer several problems. First, traditional search engines only process the text of documents but do not analyze their structures for extracting document specific information. For example, if a user is looking for some information about a chairman of Computer science department a reasonable action may be for him to type in a search engine is “computer science chair”. However, the search engines will not answer the query in the few results records. The reason is that at computer science chair websites and the department chairs are more commonly referred to as department heads. This examples show us, a search engine fail to recognize to retrieving documents in response to a query to recognize similar concepts when they are expressed using different domain. That means the search engines was unable to recognize that the words referring the same concept.

Second, traditional search engine is lack of data link from different data source. Domain specific documents can contain relations between each other that are not considered by traditional web search engines. In traditional web search engines lacking of search are occurred because of no single data source can completely satisfy the users query. For example “*find all research papers that academic have published in*”. In this example, the traditional search engine fails to find correct answer because no single data source can completely answer the information need. However with the integration of different data source semantic web can answer this question. The reason is that traditional search engine is based on keyword matching, not designed for answering structured queries.

Third, in traditional search engine once a set of relevant document is identified using matching

criteria, the user to input the document and process the information that is contained in the documents. As a consequence, when a user's need to search multiple information from different document he/she has to manually process each document from each result and merge together the relevant information by himself. In general, these studies come up with the idea to model query answering over web of data for Afaan Oromo to eliminate the problems of the existing keyword based search engine.

Motivation

Recent advances in query answering over web of data provide end user with more and more sophisticated tools for querying linked data by expression their information need in natural language [11][19]. The idea is motivate to study in the area of query answering over web of data. Here, the prime motivation of our thesis is to converting the current web dominated by unstructured and semi-structured documents into a "web of data". To allows access to the wealth of structured data available the web. In the above section (1.2), we have explained the limitations of our current search engines (which means keyword-based) that cannot extract rich information from our domain specific. Therefore, they only present very simple document information to users and not answer exact query for the user. However, our query answering system has domain specific knowledge and expects to answer user query over web of data. The information need is not sustained by the current search engines. Therefore, we believe that building query answering system over web of data can greatly increase the usefulness of the existing search engine.

1.3. Research Questions

After conducting this study the researcher must be able to answer the following research questions.

1. How to design Query Answering system over Web of Data for Afaan Oromo?
2. How can we deal with the language specific components of the query answering system?

1.4. Objectives of the Study

1.4.1. General Objective

The general objective of this study is designing a query answering system over web of data for Afaan Oromo language.

1.4.2. Specific Objective

In order to accomplish the above stated general objective, the following specific objectives are formulated.

- Review literature with regard to the problem domain.
- Identifying components of query answering system over web of data
- Designing the architecture of query answering over the web of data for Afaan Oromo
- Implementing the system prototype.
- Evaluating the system.

1.5. Scope of the Study

The scope of this thesis is to model query answering system over web of data for Afaan Oromo documents. Due to time and resource constraint, the scope of this thesis is limited to answering user query for simple sentences and evaluated on education domain ontology.

Methodology

In order to realize the objectives of our work, we have used the Design Science Research (DSR) methodology. DSR is considered as a practical problem solving technique [20][21]. However, these approaches pay little attention to the organizational context of Information Technology (IT)[22][23]. Describing IT artifacts as carriers of “social structure” and embedded in social practices. According to March and Smith[24], IT artifact can be categorized as a construct, model, method or Instantiations. Constructs define the basic concepts and language in which problems and solutions are defined and communicated. Models use constructs to represent the real-world contexts of the design problem and solution spaces. Methods define processes, such as solution algorithms. Instantiations show that constructs, models, and methods can be implemented in a working system. The DSR process model by [25] is adopted as shown as shown in figure 1.4.

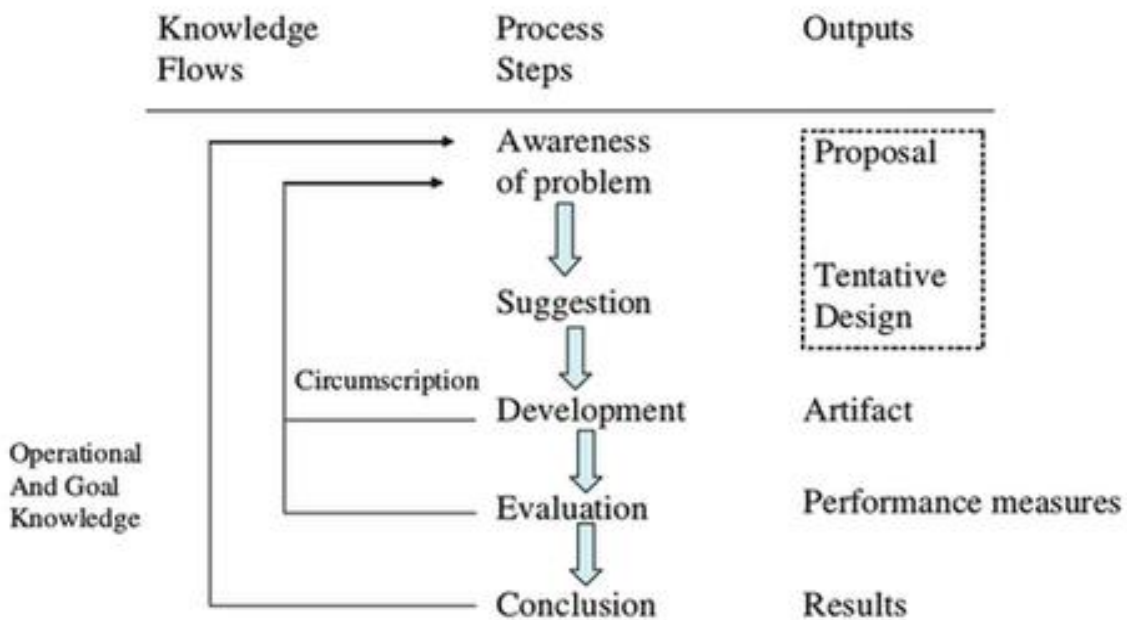


Figure 1.4: Design Science Research Framework

The first phase involves understanding the problem by providing critical literature reviews. The aim is to better understand the current approach and identifying the problem being motivated the area of semantic web search engine. At development phases the designer beginning of data collection, search for new explanatory knowledge and codification of design knowledge can begin. At third phase, artifact performance data become available. The final phase, conclude the results and categorized for future work. Further description discussed in next section from the origin to final phase.

Literature Review

For better understanding of semantic web search engine, literature reviews on different aspects will be conducted. This literature reviews give the general information for the proposed work. For this study, the most related literature reviews will be conducted to acquire enough understanding of the various components of the semantic web search engine. Specifically, literatures in the area of automatic indexing (concepts and approaches), search engine, semantic web search engine, ontologies, query answering over web of data, Afaan Oromo language writing system, information extraction methods, and existing automatic indexing researches for Afaan Oromo and other languages has been reviewed. Reading literatures and related works helps to gain the required knowledge on a certain subject, identifying the research problem and also assists in identifying the right methods and tools for implementing the different components of the system.

Data collection

For this thesis, education data written in Afaan Oromo from related research, Afaan Oromo news magazine and other website are collected. The data is collected for two categories; data to populate the ontology and data for applying user query. The first category contains a set of data which is a collection of concepts instances to be mapped to the ontology and the second category contains a set of data answering user query.

Experimentation method/Tools

In order to accomplish the objectives of our research, different methods and tools are used those are: **protégé** which is very popular tool for editing and managing ontologies. Java (Eclipse) environment to manipulate the ontology using Jena framework, RDF query language to use as a backend for Jena to store the Ontology, SPARQL query to extract related query from documents and to access the concepts in the ontology further will be discussed in chapter five.

Evaluation Method

This study is to implement query answer system to explore web of data. The system user can formulate her/his query with Afaan Oromo language. The system convert then the user query into SPARQL to interrogate Afaan Oromo web of Data, and finally the result to the user.

So, to evaluate the performance of the proposed work, we verify that the goal has been performed according to the rules and procedures that we have set in our thesis. Based on the rules and procedure, the result of the test has been evaluated by applying recall, precision and F-measure techniques. Further it will be discussed in chapter five.

1.6. Significance of the Study

With the rapid growth of documents written in Afaan Oromo language on the Internet, it become difficult to searching web documents based on their contents, rather than based on keywords as traditional search tools do. For this reason, we model query answering over web of data for Afaan Oromo with the following significance:

- Knowledge will be organized in conceptual spaces according to its meaning.
- Keyword-based search will be replaced by query answering.
- Requested knowledge will be retrieved, extracted and presented in a human friendly way.
- Query answering over several documents will be supported.

1.7. Thesis Organization

The rest of the thesis is organized as follows: Chapter 2 describes literature review of query answering over web of data, and describes related work which presents a most approach of semantic web search engine for Afaan Oromo and Non- Oromo language. Chapter 3 presents Afaan Oromo writing system. Chapter 4 describes general framework and algorithm of the thesis. Chapter 5 describes implementation and evaluation of the system and chapter 6 describes the conclusion and future work of this study.

CHAPTER TWO

2. LITERATURE REVIEW

2.1. Overview

Various techniques and approaches designed to perform Query Answering over Web of Data has been studied in different language[13][26][27]. In this chapter, we introduce literature review regards to our problem domain and related researches conducted for Afaan Oromo and other language and the last section will give over all a summary of the reviewed related works.

2.2. Web Search engine

Information Retrieval (IR) is the key technology of search engine to obtaining information from a collection of file. IR is mainly focus on retrieve information from text documents. As World Wide Web increases, IR is widely used in web search engines for retrieving information from massive heterogeneous documents.

The current web search market is dominated by several search engines like Google, Bing, Baidu etc. Though their architecture and used techniques may differ greatly in details, the basic workflow of search engines remains unchanged: crawling, indexing and searching. Here, we briefly talk about the workflow of a traditional search engine basing on the anatomy of Google in [28].

2.2.1. Web crawling

For a search engine, documents are the source of information and will be preceded several times. An important component, crawler, is specifically designed to fetch documents. There are different kinds of crawlers:

For a general purpose web search engine, given some initial links, the crawler iteratively downloads document contents and follows discovered links in pages [28]. For a domain specific search engine, the crawler should only download documents that are about its interested topics

[29]. For an enterprise search, the crawler should not only download all internal documents by links, but also need to scan folders to find different files[30].

Implementing a reliable and fast web crawler for large scale web is not trivial as it may seem. As the web has more than billions of web pages, a web crawler should be designed to scale well in such environment. To achieve fast crawling, Google implemented a distributed crawling system, in which workloads are distributed among several machines and each crawler maintains connections to hundreds of web hosts in parallel. In order to reduce DNS lookup time and URL fetching time, each crawler maintains a DNS cache and several crawlers are served by a dedicated URL server. Reliability is also an important issue to consider, since formats and structures of fetched documents are arbitrary and the attempt to parse them may even crash the whole system.

2.2.2. Indexing Documents

Usually, a downloaded document is not plain text but has other formats, e.g. HTML, PDF, Word etc. These files have to be parsed to get their plain text and the content and metadata (e.g. title, keywords, and author) of processed files are stored in a file database.

To make documents searchable, its plain text needs to be tokenized, i.e. extracting separated words. Tokenizing can be simply based on splitting words by whitespace, but can also involve many complex techniques, including removing stop words, splitting words by particular symbols, word stemming etc.

For fast search, inverted index are built for the whole document set. With inverted index, the documents that a word occurs can be directly retrieved.

2.2.3. Search for documents

For several keywords from users, a search engine retrieves a list of relevant documents and ranks them according to specific metrics of relevance measurement. Search engines find documents by checking the inverted index. Documents contain all keywords are returned. After fetching relevant documents, search engines rank documents. Ideally, documents should be ranked in

descending order of their relevance.

Measuring document's relevance to keywords is complicated. The most widely used method is term frequency-inverse document frequency (tf-idf), which counts a word's occurrence in a single document and in the whole document collection. While Google proposed PageRank, an algorithm that calculates the relative importance of web pages without concerning keywords. Here we first explain the two most important techniques and then discusses some other commonly used techniques.

i) tf-idf

tf-idf is one of the most commonly used algorithms in statistics to evaluate how important a word is for a set of documents. It consists of two parts: term frequency (tf) and inverse document frequency (idf).

tf: tf is about the importance of a word in a document. Intuitively, the more a word appears in a document, it is considered to be more important in the document. Therefore, the simplest way to calculate a word's tf is to count its number of occurrence in the document. However, a problem of this approach is that it has a bias towards long documents, i.e. words in longer documents tend to get higher scores. A revised way is to normalize the score by dividing raw tf with the highest tf of a word in the document, as the following formula shows.

$$tf(t, d) = \frac{f(t, d)}{\max\{f(w, d) : w \in d\}} \quad (2.1)$$

Here t denotes a specific term and d denote a document.

idf: idf measures the rareness of a word among a collection of documents. If a word appears in a lot of documents, it seems to be a general term rather than a meaningful word representing user's searching intent. The way to calculate idf is as follows, where a rarer word gets a higher score:

$$idf(t, D) = \log \frac{|D|}{|\{d \in D: t \in d\}|} \quad (2.2)$$

Here, t denotes a term and D denotes the collection of documents.

Finally, a tf-idf score is calculated as:

$$tf - idf(t, d) = tf(t, d) * idf(t, D) \quad (2.3)$$

To gain a high tf-idf score in a document, a word should be a rare word in the whole document set but should occur many times in that document. In return, this implies the documents' high relevance to that word. In practice, if a user searches for some keywords, search engines calculate the tf-idf score of all documents containing these words. Retrieved documents are ranked according to their tf-idf score in descending order.

ii) **PageRank**

PageRank is an algorithm to measure the relative importance of web pages. Most web documents have hyperlinks between each other. In the PageRank algorithm, the PageRank score measures the importance of a page; each web page initially has an equal PageRank score. A hyperlink from it to another page counts as a vote for that page. The following formula shows the calculation, in which R denotes PageRank score, N denotes the total number of out links of a page and B denotes all backlinks of a page:

$$R(u) = c \sum_{u \in B_u} \left(\frac{R(u)}{N_u} \right) \quad (2.4)$$

The intuition of the algorithm comes from the behavior of a web surfer who randomly surf the Internet by clicking on links of pages. An illustration of PageRank is shown in figure 2.1. However, if a web page has no outgoing links, apparently users will not stay on the web page forever. By adding a damping factor to the algorithm the web pages will keep some fraction of their PageRank score when voting for others. This avoids the infinite growing on the PageRank

value for some pages that has no outgoing links. In general, this model successfully ranks web pages without considering complex factors and provides accurate results.

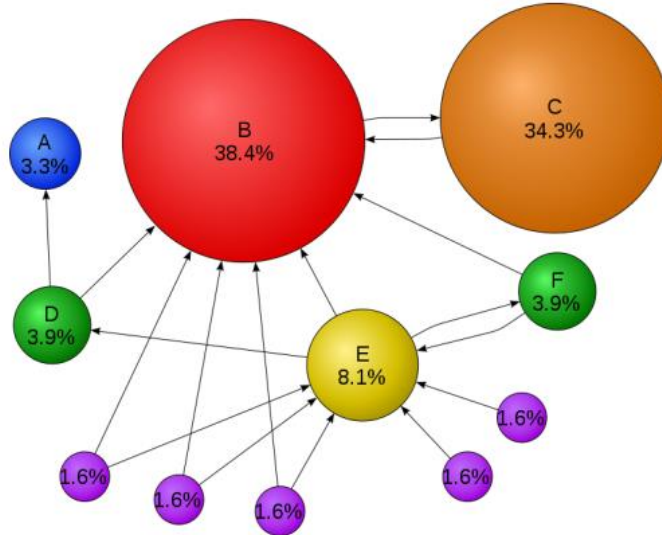


Figure 2.1: An illustration of PageRank

2.3. Semantic web Based Approaches

Existing web contains billions of documents and has many administrative limitations. To solve this problem, Tim Berners-Lee, inventor of the Web, introduced the semantic web as a conceptual model of web that makes the contents to be read and used by human and intelligent by machines. Tim Berners-Lee [5] proposed the concept of semantic web as follows:

“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”. Since, the current web depends on visual representation of information through HTML tags. This visual representation makes information clear for humans to understand but very difficult for machines to understand and process. Adding semantics to the Web comprises allowing documents which have information in machine readable forms, and allowing links to be

created with relationship values[31].

So, semantic web can enhance the currently existing Web by the interposition of a machine interpretable layer that holds the metadata of the Web document. These metadata will permit computer software to know what the Web page is about, and hence draw conclusions about it. The focus of semantic web is to share data instead of documents .This revolutionary advance can be utilized by software agents, so the information can easily be found, shared, integrated, exchanged and reused on the Web and across application, enterprise, and community boundaries[32].

2.3.1. The structure of the Semantic Web

The Semantic Web combines several existing technologies to convert the World Wide Web from a web of document to a web of data. Tim Berners-Lee [4] proposes a layered approach for achieving the Semantic Web. It is a collaborative effort led by World Wide Web Consortium based on a layered set of standards, as shown in Figure 2.2

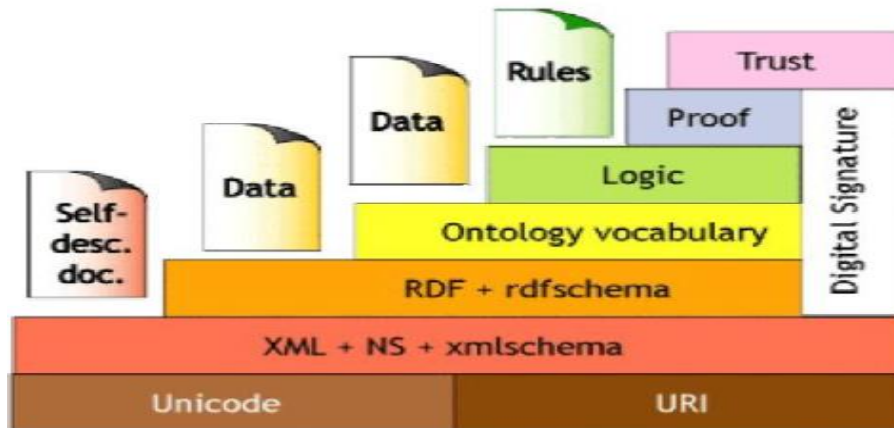


Figure 2.2: The layered structure of the Semantic web

As the figure 2.2 shows, the Semantic Web consists seven layers. The brief introduction of each layer is as follows.

i) URI

A universal resource identifier (URI) is a formatted string that serves as a means of identifying abstract or physical resource. A URI can be further classified as a locator, a name, or both. Uniform resource locator (URL) refers to the subset of URI that identifies resources via a representation of their primary access mechanism. A uniform resource name (URN) refers to the subset of URI that is required to remain globally unique and persistent even when the resource ceases to exist or becomes unavailable. For example, The URL “<http://www.semanticweb.org/alex/uni>” identifies the location from where a Web page can be retrieved.

ii) Unicode

Unicode provides a unique number for every character, independently of the essential platform, program, or language. Before the creation of Unicode, there were several encoding systems. The various encoding made the use of data difficult. Any given computer needed to support many different encodings. There was always the risk of encoding conflict, because two encodings could use the same number for two different characters, or use different numbers for the same character. Examples of older and well known encoding systems include ASCII and EBCDIC.

iii) XML and XML Namespace

With XML (extensible markup language) namespace and XML schema definitions check that there is a common syntax used in the semantic Web. XML namespaces allow identifying different markup vocabularies in one XML document. XML schema assists for expressing schema definition of a particular XML document.

iv) RDF and RDF Schema

RDF is represented based on triples O-A-V that form a graph data with a relation among object (a resource), an attribute (a property), and a value (a resource). RDF Schema (RDFS) defines the

vocabulary of RDF (resource description Framework) model. It delivers a mechanism to describe domain-specific properties and classes of resources to which those properties can be applied, using a set of basic modeling primitives (class, subclass-of, property, subproperty-of, domain, range, type). However, RDFS is rather simple and it still does not provide exact semantics of a domain.

v) Ontology

Ontology involves a set of knowledge terms, including the semantic the interconnections, vocabulary, simple rules of inference and logic for some particular topic. Ontologies facilitate knowledge sharing and provide reusable Web contents, Web services, and applications. Few of basic ontology languages are DAML (DARPA Agent Markup Language), , OWL (Web Ontology Language) and OIL (Ontology Interference Layer). OWL is developed starting from description logic and DAML+OIL. Ontology web language (OWL) is a set of XML elements and attributes, with well-defined meaning, that are used to express terms and their relationships (e.g. Class, equivalentProperty, intersectionOF, unionOF, etc.).

Ontology web language (OWL) elements extend the set of RDF and RDFS elements, and the OWL namespace is used to denote OWL encoding. OWL comes in three classes – OWL Lite for taxonomies and simple constraints, OWL DL for full description logic support and OWL full for syntactic freedom of RDF. Ontology web language is broadly used for ontology representation. In practice, ontologies are often developed using integrated, graphical, ontology authoring tools, such as OILed, OntoEdit and Protégé. Protégé enables extensible infrastructure and allows an easy construction of knowledge rich domain ontologies.

vi) Logic, Proof, Trust and Digital Signature

The last three layers are described as follows; the logic layer is used to develop the ontology language more and to permit the writing of application-specific declarative knowledge. The proof layer comprises the actual deductive process as well as the representation of proofs in Web languages and proof validation. Finally, the Trust layer will appear through the use of digital

signatures and other kinds of knowledge, based on recommendations by trusted agents.

For the semantic Web to become more expressive enough to help in a wide range of situations, it will become essential to construct a powerful logic language for making inferences. The next step in the architecture is ‘Trust’ and ‘Proof’. No more described about these layers though they will become important in future.

2.3.2. The Semantic web application area

Though not fully developed, the Semantic Web has been applied to various areas:

Knowledge management: With semantic metadata, software can better manage documents. Knowledge management software can either be built for internal environments or for public usage, like a search engine. **FreeBase1**, **DBPedia2** are knowledge bases that use ontology to present their data and user can retrieve the full knowledge graph of their searching entity.

Personal agents: A personal agent is software that collects web content from diverse sources, understands their semantic meaning and finds accurate answers. **Evi3** and the well-known **Siri** are examples of these kinds of software.

2.3.3. Challenges for Semantic web Search Engine

The Semantic Web search engine depends on the ability to associate formal meaning with content. Conversely, the nature of the Web challenges many of the assumptions of traditional knowledge representation work, and requires us to look at the problem from a new perspective. The impacts of some of the most significant characteristics of the Web are discussed below:

The Web is distributed: One of the factors in the proliferation of the Web is the freedom from a centralized authority. Conversely, the Web is the product of many individuals; the lack of central control presents many tasks for reasoning with its information. First, different communities will use different vocabularies, resulting in problems of synonymy (when two different words have the same meaning) and polysemy (when the same word is used with different meanings). Second, the lack of quality control means that each page’s reliability must be questioned. An

intelligent web agent simply cannot assume that all of the information it collects is correct. Furthermore, since there can be no global enforcement of integrity constraints on the Web, information from different sources may be in conflict. Some of these conflicts may be due to philosophical disagreement; different political groups or nationalities may have fundamental differences in opinion that will never be fixed.

The Web is dynamic. The web changes at an improbable speed, much faster than a user or even an intelligent web agent can keep up with. While new pages are being added to the web, the content of existing pages is changing. Some web pages are fairly static, others change on a regular root and still others change at unpredictable intervals. These changes may vary in significance: although the addition of punctuation, correction of spelling errors, or reordering of a paragraph does not affect the semantic content of a document; other changes may totally alter meaning, or even remove large amounts of information. The rapid pace of information change on the Internet poses an additional challenge to any attempt to create standard vocabularies and provide formal semantic web.

The Web is massive. In the year, estimates placed the number of index able web pages at over 2 billion. Even if each page contained only a single piece of agent-gatherable knowledge, the collective database would be large enough to take most reasoning systems to their knees. To scale to the scope of the ever growing Web, we must either limit the expressivity of our representation language or use incomplete reasoning algorithms.

The Web is an open world. A web agent is not free to assume it has gathered variety of all available knowledge; in fact, in most cases an agent should assume it has gathered rather little available knowledge. Even the largest available search engines have only crawled about 25% pages. However, in order to deduce more facts, many reasoning systems make the closed-world assumption. That is, they assume that anything not entailed in the knowledge base is not true. Yet it is clear that the size and evolving nature of the Web makes it improbable that any knowledge base attempting to describe it could ever be complete.

2.4. Ontology Based Approaches

According to the history of artificial intelligence shows the knowledge is critical for intelligent systems. In many cases, better knowledge can be more significant for solving a task than better algorithms. To have truly intelligent systems, knowledge needs to be processed, captured, reused and communicated. Ontologies support all these tasks[32][33]. Currently, Ontology is becoming very important because we have a lack of standards (shared knowledge) which are rich in semantics that difficult to represented in machine understandable form[34]. Introductory descriptions on Knowledge Representation, Ontology to represent knowledge, basic ontology component, Building Ontology search and ontology languages and tools are discussed below.

2.4.1. Knowledge Representation (KR)

Several of the problems with processing and integrating XML documents could be answered if we could associate machine understandable meaning with the tags. This meaning could be used to convert from one DTD to another, or reason about the consequences of a given set of facts. According to[35], Knowledge representation, an important sub-field of artificial intelligence, can provide insights into these problems. A knowledge representation scheme defines how a program can model what it knows about the world. The goal of knowledge representation is to create schemes that allow information to be efficiently stored, modified, and reasoned with.

2.4.2. Ontology to represent knowledge

The term “**Ontology**” originates from the discourse of philosophy. In philosophy, it means the nature of existence. In our context, ontology denotes as an explicit specification of conceptualization used to help programs and humans share knowledge, as Thomas R. Gruber proposed in [36]. From the same perspective, philosophers have different senses for the term Ontology and ontology [37]. Ontology is uncountable having no plural form, is addressed by Aristotle, and refers to a discipline that studies the nature of being or theory of existence. It is intended to give answers for questions like “What is being?” and “What characteristics do all beings have in common?”[38], Whereas ontology has a plural form and refers to a system of certain categories responsible for designing some view of the world and the system is

independent of the language even if it is reliant on a particular philosophical view[37].

Ontologies capture the structure of the domain. i.e conceptualization. This includes the model of the domain with possible boundaries. The conceptualization describes only knowledge about the domain. In other words, the conceptualization is not changing or is very rarely. Ontology is then specification of this conceptualization. The conceptualization is identified by using particular modeling language and particular terms.

Like an XML, DTD or XML Schema, ontology can provide a standard vocabulary for a problem domain. However, ontology can also contain structures or axioms that define the semantics of the vocabulary terms. These semantics can be used to infer information based on background knowledge of the domain and to integrate data sources from different domains. The documents having metadata are annotated by using ontology [39]. In this work, the main aim of ontology is to provide knowledge about specific domains that are understandable by both the computers and developers.

2.5. Basic components of ontology

To modeling the language, word can be having different structure and the components may be named differently from one ontology to another. However, the basic components are believed to be the same. These components can be categorized in to two groups; those which describe the entities of that particular domain like concepts, instances, and relationships and those that are used to describe the ontology itself. The core components are discussed below:

Concepts also called Classes: are the core components of most ontology. Concept is a unit of knowledge which is represented by a descriptive statement or a formal expression and its meaning is shared among identified group of responsible persons for the concept's domain [40]. Concept is a collection or types of objects labeled with terms. One concept can be a sub-concept of (also known as sub-class of/kind-of/part-of) another concept. This structure of concepts is usually referred to as concept taxonomies. If a concept c1 is a sub-concept of a concept c2, then the individuals of c1 are also the individuals of c2. As there are relationships among entities in

the real world, there are relationships among concepts of a particular domain as well.

Instances also called individuals or particulars: are concrete examples of concepts in a domain of interest. Instances represent specific elements attached to a specific concept in the domain ontology. Instances are the 'things' represented by a concept. Instances relate with other instances by the relationships shared by the concepts the instances belong to.

Relation: refers to interactions between concepts or concepts' properties. Relations in ontology describe the way in which individuals relate to each other. Relation can be defined directly between instances or between concepts. Relations defined between concepts describe the relationship between all instances of these concepts.

Axioms are explicit rules defined to constrain the use of concepts and the values for classes or instances. They are used to model sentences that are always true.

2.6. Building Ontology search

As we discussed in section (2.2), traditional search is based on keyword matching, which can hardly catch the actual conceptualization associated with user's interests. There are research efforts trying to aid search with ontologies to improve the search accuracy. Several procedures exist in integrating ontology search: input analysis and translation[41][42], Ontology annotation of documents and various ontology exploration techniques[43][44]. In our context, we only discuss ontology annotation and ontology search based on semantic web search engine.

2.7. Ontology languages and tools

There various formal languages used to construct ontologies. The basic ontology languages tools such as: Web Ontology language (OWL), Onto UML, OntoEdit, WebODE, and Protégé are some of the most known ontology development tools. In our thesis, we use OWL to annotate our metadata and protégé is selected to building the education domain ontology. Protégé is chosen because it is free, easy to use, and it supports the OWL language as well, and has all the services needed to construct this particular ontology. Here we introduce the basic syntax of OWL and

related tools for aiding the development and execution of OWL[6].

Resource Description Framework (RDF)

RDF stands for resource description framework. According to W3C RDF is a standard model for data interchange on the semantic Web [18]. The Resource Description framework (RDF) has features that facilitate data integration even if the underlying schemas vary, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed.”

The resource description framework (RDF) is an XML-based language for describing information contained in the web resource. A resource can be web-page an entire web site or any item on the web that contains information in some form. With Resource Description framework it is possible to add predefined modeling primitives for expressing semantics web of data to a document without creating any conventions about the structure of the document[45]. The fundamental concepts of RDF are resources, properties and statements discussed as follows.

Resources: A resource as an object can be a thing such as person, a song, or a Web page. Every resource has URI to identify itself.

Properties: They are a special kind of resources that describe relations between resources, for example “written by”, “title” ,“age” and so on.

Statements: Statements assert the properties of resources. A statement is object attribute-value (O-A-V) tripe, consisting of a resource, a property and a value. Value can be either being resources or literals. An RDF is mostly expressed using something known as a triple which is the most basic unit of information. A triple contains a subject, a predicate and an object. Namespaces are provided to solve the problem on ambiguity in RDF [12].

The idea of this simple unit of information is that it could be expressed in various formats that could be easily read by machines. The RDFS (RDF Schema) is used for describing the properties and classes of an RDF document. RDFS acts similar to the function of metadata for RDF.

Web Ontology Language (OWL)

Another standard from the W3C Consortium to aid in the progress of Semantic Web is Web Ontology Language (OWL). OWL is the standardized and broadly accepted ontology language for the semantic web. It is developed on top of RDF and RDF Schema (RDFS) and also uses XML-based syntax. Except for the simple hierarchical relations of RDFS (Class, SubClassOf, property Of, SubPropertyOf), OWL provides more expressive syntactic axioms, which greatly enhances the ability of RDF to model complex relations.

In our context, OWL is used as the storing format of metadata. The rich syntactic elements of OWL are not fully used. However, concerning that the semantic web is becoming more popular and OWL is also widely used as the metadata markup language. It is used for the sake of high interoperability with other Ontology-consuming programs and further extensibility[46].

SPARQL

SPARQL is another W3C standard in the category for Semantic Web. It is a query language similar to that of Structured Query Language (SQL) for Relational Database Management Systems (RDBMS). SPARQL is a query language for retrieving and manipulating data stored in RDF format. It is a standard language for accessing RDF and OWL. RDF data are stored in the form of triples and a SPARQL query is a cascading list of triple queries[46][45].

Ontology tools

In order to create ontologies, various ontology editing tools are available on the web. All these tools are efficient enough to create and manipulate ontologies for a specified domain. OntoEdit ,Protégé, Hozo, OilEd, WebOnto, Swoop, OntoLingua, Top Braid Composer, etc. are few names of available ontology editors. Each of these tools has its own features and performance levels based on which we pick the latest version[47][48].

2.8. Related Work

There is various literatures review that has been done on area of Query Answering over Web of Data. However, it doesn't considering for Afaan Oromo language. In this section some of related work has been discussed as follows.

In 2013, Tesfaye Guta [12] proposed a search engine for Afaan Oromo texts that enables searching documents on the web. According to this work, the mainly focus on three components crawler, Indexer, and query engine that are optimized for Afaan Oromo. First, Crawler downloads documents and then filtering of these documents. Next, document that are identified as Afaan Oromo are preprocessing and stored in an index for later retrieval. Finally, queries supplied checked for a match in the index and matched documents are displayed through an interface in a ranked order.

In this study, the indexer components, normalizer, stopword removal, stemmer and Lucene are discussed in details. The indexing part of the system is done by using Lucene indexer. Lucene uses idf (Inverse document frequency) approach for indexing. IDF is a keyword based method to represent a document. These indicate that they need semantic search for Afaan Oromo search engine.

Birhanu Anbase (2017) proposed “Developing Latent Semantic Indexing based Information retrieval for Afaan Oromo”[13]. This research is conducted to investigate the advantage of Latent Semantic Indexing (LSI) techniques to build semantic indexer based on information retrieval for Afaan Oromo .The main task of this study is to extract concepts from a given corpus by looking for words that co-occur frequently without giving emphasis to the relationships between the concepts. For this study, the potential of LSI approach in Afaan Oromo text retrieval is investigated. To test the approaches, 70 Afaan Oromo documents and 9 queries were used. The performance of the system after user relevance feedback is measured by using recall and precision. The experiment shows that the performance of the prototype on the average registered 67% precision and 63% recall. This implies that it is possible to improve the performance by designing good stemmer and standard corpus for Afaan Oromo languages. The proposed system

is implemented with appropriate tools. Those appropriate tools are; Java (Net beans 8.0.1) which runs the prepared corpus and JAMA (Java Matric) package used for generating term document matrix and for constructing and manipulating real, dense metrics.

Even though, these approach have a lot of limitations which affect the performance of the retrieval system when we compare with semantic web search engine. The LSI method has a lot of problem of dealing to responding indirect queries; if the words in the query do not exist in any of the documents, then it may generate irrelevant documents given with high rank and miss the relevant ones or give the relevant ones lower rank. LSI that takes a query as pseudo-documents and looks for documents very similar to it, however these approaches cannot retrieve documents which is related to the give queries, also LSI based concept Indexer ignore the relationship between concepts. Furthermore, as the size of documents increases the performance of the indexer degrades and time intensive.

Fekade and Genet (2015) presents an approach to build Amharic semantic search engine documents using Ontology concept[26]. This paper working on semantic search engine that dedicated knowledge base organized systematically enters of concepts and associated relationships between documents. The main goal of this study is to design Amharic Semantic search engine which returns list of Amharic documents very similar to the user query either directly or indirectly.

The proposed systems mainly focus into the specific features of Amharic language. It is composed of four major components, crawler, Query Interface, concept Indexer, Ontology and query processor. The descriptions of those components work as follows: Crawler- the main task of crawler is to extracts URL and content of Amharic documents from a given page and stores them in document repository. The crawler uses http protocol and multiple threads that process concurrently to fetch pages. Concept-Indexer- extracts unstructured Amharic documents with semantic information concepts and associate semantic information with weights. Document annotation and Indexer has the main components of concept Indexer. At this step, word to concept mapping of predefine rules and looking for the best in the knowledge based concept

identification are used. Ontology- provides shared concepts sport which built manually by language and sport domains experts. In this work, the similarity between concepts is measured by using the popular distance based approaches presented in[49]. Query processor- is responsible to accepts user query, annotate it using semantic knowledge, retrieve documents and rank them.

In order to validate the performance of the proposed work this study compare with the result of classical keyword-based Amharic search engine and evaluated by popular information retrieval base relevance Recall, Precision and F-value. However, this approaches are restricted to Amharic languages, hence do not address basic characteristic of Afaan Oromo language.

Raj, Syam, and S. Sarumathi presents Ontology based semantic search engine [50]. The idea of the paper is to analyze the knowledge about real world and to create standard stabled rules and relation types to translate human (natural) language in a machine. In the WWW information is presented in natural language which is not rich enough to convey formal meaning which means not machine process-able. The current web is a web of document and understandable only to humans. This makes information retrieval process very hard. Trying to make machines act as human is a very complex task. This paper solves such kinds of problem that content on the current web is intended only for human consumption. In this paper medical knowledge about cancer is combined with the semantic web search engines to build an ontological structure. The implementation and learning process is achieved by comparing with some knowledge organization system. The knowledge acquisition in semantic web is done by RDF explorer. RDF scheme defines relationship and those relationships make the searching on a different level. In this work protégé tools are used which helps to create the ontology.

M. Kayed, A. Sayed, and A.L. Muqrishi in [51] presented CASENG: Arabic Semantic Search Engine and proposes a framework that facilitates an effective search over the semantic web. Firstly the figure out various factors that influence the search experience over the Internet for Arabic language since Arabic language is it is highly inflectional and derivational language. Because of this reason current web such as Google, Gmail, yahoo and others searching on the web based on ranking algorithm or number of terms is not gives an efficient solution. Secondly

the semantic web technologies necessary to perform a basic search over the Internet will be described-that is Resource Description frame work (RDF), RDF Schema, web Ontology language (OWL), RDF Query language and Lucene that is the open source Java library used for indexing and searching. Thirdly the distinguish search engines which supports for Arabic language. Finally CASEng makes use of Lucene spell checker in order to correcting query spelling for Arabic queries. It is built-based on N-gram Distance and Levenstein Distance.

For this study two Lucene spell checker indexing techniques are used. The first indexing techniques are called “indexing a field” (IFe) in which the indexing mechanism is done term by term. Thus, spelling correction is done term by term. The second techniques is called “indexing file”(IFi) in which the spell checker checks the query and suggests the similarity line by line. As the experimental results shows, Lucene spell checker performs well when the IFI indexing techniques is applied to measure the performance of the proposed search engine. In general the proposed system emphasized on natural language understanding approach which is considered as part of semantic web.

In 2017, Ali A, Feras K, and Khaled S in [1] presents an Arabic QAS using ontology based on the domain knowledge or ontology in order to answer natural language query. Prior to the implementation, it was key to perform some natural language processing (NLP) tasks this assisted in analyzing the questions such as normalization, tokenizing, stop word removal, stemming and tagging. Furthermore, this study present how to develop the ontology through the protégé tool, how to translate the inquiries into triple patterns and build the SPARQL queries which are the mechanism to retrieve the answer from knowledge graph. The proposed system has achieved promising results with accuracy of 81%, which provides an important suggestion for further in-depth study and analysis in this area.

Dennis D, Andreas B, Kamal S. and Pierre M in [52] presents new approach “Towards a Question Answering System over the Semantic Web” for translating natural language query into SPARQL. The main idea of this study is to query several KBs simultaneously in different languages and can easily be ported to other KBs and language. For the evaluation purpose they

showed using five different well known and large KBs; DBpedia, Wikidata, MusicBrainz, DBLP and Freebase as well as 5 different languages namely English, German, French, Italian and Spanish are used. In addition to this, how to integrate different data source to make it easily accessible by the end users has been discussed. In general, this study provided a conception solution for multilingual, KB-agnostic Question Answering over the Semantic Web.

2.8.1. Summary

In this section a numbers of researches related to semantic web search engine has been reviewed that focused both on local and foreign language. Moreover, table 2.1 clearly shows summary of related works of this thesis based on their language, approach and limitations.

Table 2-1: Summary of related works

Language	Research Title	Approach	Limitations
Afaan Oromo	Afaan Oromo Search Engine[12]	Keyword based	Search irrelevant queries
	Developing Latent Semantic Indexing based Information retrieval for Afaan Oromo [13]	Corpus based	-Time intensive -Incapability of handling indirect Queries
Amharic	Towards Amharic semantic search engine[26]	Ontology based	Classic based search engine Approaches are not consider Afaan Oromo language
English	Ontology-based semantic search engine[50]		
Arabic	CASENG: Arabic Semantic Search Engine[51]		
Multiple language	Towards a QA System over the Semantic Web[1]	Large KB (DBPedia, free base)	Depends only structured data

CHAPTER THREE

3. Overview of Afaan Oromo

In this section, we are briefly discussed Afaan Oromo writing system (the alphabets, the punctuations, and Abbreviations), Word Categories, Morphology, and Questions in Afaan Oromo.

Afaan Oromo or the Oromo language belongs to the Cushitic branch of Afro Asiatic language family and one of the languages of Lowland groups within the East Cushitic group. It is the third most widely spoken language in Africa, after Hausa and Arabic [53]. Currently, it is an official language of Oromia Regional State which is the biggest region among the current Federal States in Ethiopia. Among the major languages that are widely spoken and used in Ethiopia, Afaan Oromo has the largest speakers[54]. It is a widely used as both written and spoken language in Ethiopia[55]. According to (Ibrahim, 2015) Afaan Oromo is purely natural nature based language. Each and every roots of Afaan Oromo were created from either corresponding Sounds or available roots and thus, converges to sounds proximate to it[56].

3.1. Afaan Oromo Writing System

Writing system is a conservative method of visually representing verbal communication. With regard to the writing system, Qubee (a Latin-based alphabet) has been adopted and become the official script of Afaan Oromo from 1991[53][56]. This writing system was adopted from the fact that its characters explicitly represent the vowels and the consonants of the language.

3.1.1. Afaan Oromo alphabets

The Qubee writing system of Afaan Oromo has a total of 33 letters that consists of all the 26 English letters (a...z) and the 7 combined consonant letters (ch, dh, sh, ny, ph, ts, zh)[56]. Like English alphabet, the Afaan Oromo alphabet characterized by capital and small. In Afaan Oromo language, as in English language, the vowels are sound makers and are sound by themselves. Vowels in Afaan Oromo are characterized as short and long vowels.

All the vowels in English (a, e, i, o and u) are also vowels in Qubee Afaan Oromo[57]. They have two natures in the writing system of Afaan Oromoo language and results in different meanings. A vowel is said to be short, if it is one in number and long vowel, if it is two which is the maximum. Example: *laga (river)*, *Bona (summer)*, are short vowels, whereas *laagaa (throat)*, *Boonaa (pride)*, are long vowels. The rest of Qubee Afaan Oromoo is consonants. The combined consonant letters are known as “*qubee dacha*” which was listed above. Doubling of a consonant is a phonemic in Afaan Oromoo. Example: *Damma (honey)*, *Callaa (product)*, *Ganna (winter)*. But “*h*” character is not geminated at all.

3.1.2. Afaan Oromo Punctuation Marks

Punctuation is placed in text to make meaning clear and reading easier. Similar to English and other Ethiopian language, Afaan Oromo punctuation marks follow the same punctuation pattern used in English and other languages for writing system purpose[53]. For example:

Qooduu Comma (,) -is used to separate listing of ideas, concepts, names, items, etc.

Tuqaa Full stop (.) -is used at the end of a sentence and in abbreviations.

Mallattoo Gaafii Question mark (?)-is used in interrogative or at the end of a direct question.

Rajeffannoo Exclamation mark (!): is used at the end of command and exclamatory sentences.

Tuqlamee colon (:): is used to separate and introduce lists, clauses, and quotations, along with several conventional uses, and etc.

3.1.3. Afaan Oromo Abbreviations

Abbreviations are mostly formed by taking initial letters of multiword sequences to make up a new word. In Afaan Oromo, abbreviations are used to represent dates A.L.I (Akka Lakkoofsa Itiyooophiyaa) to mean in Ethiopian calendar. Moreover, personal titles can be abbreviated like that of English language. For examples: “Aadde” is abbreviated as

“Aadd.”(Mrs.), Obbo is abbreviated as “Obb.”(Mr.). Organizations names are also abbreviated. For example, “M/Murti” (Mana Murtii) (Court).

3.2. Word Categories of Afaan Oromo

Words are the basic part of any given language. The arrangement of word or their combination depends on the rule or grammar of that language. The combination of these words on the bases of the language gives us sentences. The meanings of these sentences depend on each word of the sentence and the way they are arranged. However, the extent to which a given word determines the meaning of a sentence depends on the contribution of that word. All words do not have equal contributions to sentence meaning. Their contribution depends on their category and their feature. In general, word categories can be identified by looking at the meaning (semantic) of that word, by looking at the form (morphology) of that word or by looking at the actual position (syntax) of that word[57].

In this section, we have been discussed the basic things of the word categories that are used in Afaan Oromo language and have contributions to our study. According to [58] , Afaan Oromo words are categorized into eight grammatical categories. These are maqaa (noun), bamaqaa (proper noun), xumura (verb), ibsa xumuraa (adverb), ibsa maqaa (adjective), qabsiistota (conjunctions), maxxantoota (affixes) and Interjection. The details description as follows:

Noun (Maqaa)

Nouns are names that are used to name or identify things, people, animals, places or abstract ideas. In Afaan Oromo most of the time a sentence begins with a noun which starts with capital letter and it uses a noun as a subject followed with subject markers (-ni, or -n.). For instance, words that are bold in the following sentences are nouns.

Fardi marga dheeda. (The hourse grazes grass).

Keelloon filannoo kooti. (Yellow is my preference).

Proper noun (Bamaqaa)

Noun can further be classified as common or proper noun. A proper noun is a noun that will

name a specific, usually a one-of-a-kind item. In Afan Oromo, it begins with a capital letter no matter where it occurs in a sentence.

Tola barsiisaa dha. (Tola is a teacher).

Verb (xumura)

Verb is the most important part of a sentence that says something about the subject of a sentence, expresses an actions, events or states of being. In Afaan Oromo verb occurs in the final positions of a sentence as shown below.

Deebii lukkuu bite .	(Debi buy hen)
Siifan dhufte .	(Sifen has come)

Adverb (ibsa xumuraa)

Adverbs are words which are used to modify a verb, an adjective, another adverb, or a clause. Adverbs usually precede the verbs they modify or describe. An adverb indicates time, manner, place, cause, or degree and answers questions such as ‘how?’, ‘when?’, ‘where?’, and ‘how much?’. In the following examples, each of the bold words is an adverb:

Oboleetti koo **boru** deemti. (My sister will leave tomorrow.) Boru (tomorrow) is an adverb.

Adjective (ibsa maqaa)

An adjective modifies a noun or a pronoun by describing, identifying, or quantifying words. In Afaan Oromo an adjective usually follows the noun or the pronoun which it modifies. Some of Afaan Oromo adjectives are: hedduu, mara, kam, adii, qalla, tokko, kee, etc. For instance, in (Tolaan hoolaa adii bite) “*Tola bought white sheep*” the adjective **adii** comes after the noun **hoolaa**.

Conjunctions (wal-qabsiistota)

Conjunctions are unchanging words which coordinate sentences or single parts of a sentence. The main functions of conjunctions are identified as: the function of coordinating clauses

(coordination), the function of coordinating parts of sentence (coordination) and the function of coordinating syntactical unequal clauses (subordination). On the other hand, with regard to their form we can subdivide the conjunctions of Afaan Oromo into:

i. Independent Conjunctions:

a. Coordinating

Example: **garuu** – *but*

Hoolaan garuu rooba hin sodaattu. *But the sheep is not afraid of rain.*

b. subordinating

Example: akka - *that, as if, as whether*

Maaliif akka yaada dhuunfaa yookaan yaada haqaa akka ta'e adda baasii barreessi.

Write separately why it is an individual opinion or that it is an opinion about justice

ii. Suffixed Conjunctions

Example: **-f/ -fi/ -dhaaf** - *and, that, in order to, because, for*

Loon horsiisuuf bittee? - *Did you buy the cattle for breeding?*

iii. Conjunction consisting of one, two or more parts

Conjunctions consisting of two parts can be formed by two independent words or two enclitics or one independent word plus enclitic. They can be formed made up of two single conjunctions that are used after each other in order to give more detailed information about the logical relation or to intensify it.

Example: **akkam, akka** - *how, that*

Dura namni tokko beekumsa mammaaksaa akkam akka jabeeffatu ilaaluu nu barbaachisa. (*At first we have to see how a person extends the knowledge of proverbs*)

Pronoun

Like English, in Afaan Oromo **pronoun** can replace a noun or another pronoun. Pronouns are marked for number and gender. For example, pronouns like "ishee/isii" which means "she" is

feminine (singular), "isa" which means 'he' is masculine (singular), "isaan" which means 'they' is plural and can be masculine or feminine and "nuyi" which means "we" is plural and can be masculine or feminine. We use pronouns to make sentences less cumbersome and less repetitive.

Grammarians classify pronouns based on their functions and meanings in the sentence into several types, including the personal pronoun, the demonstrative pronoun, the interrogative pronoun, the indefinite pronoun, the relative pronoun, the reflexive pronoun, and the intensive pronoun.

Table 3-1: Personal Pronouns of Afaan Oromo

	1st person	2nd person	3rd person
Singular	Ani(I)	Ati(You)	Isa(He)/Ishee(She)
Plural	Nuti(We)	Isin (you)	Isaan(They)

3.3. Afaan Oromo Morphology

Morphology is a branch of linguistics that studies patterns of word formation across languages and the study of internal structure of words. Similar to other language, Afaan Oromo has a very complex and rich morphology[59]. It has the basic features of agglutinative languages involving very extensive inflectional and derivational morphological processes. In agglutinative languages like Afaan Oromo, most of the grammatical information is conveyed through affixes, (that is, prefixes and suffixes) attached to the root or stem of words. Although Afaan Oromo words have some prefixes and infixes, suffixes are the predominant morphological features in the language. Almost all Afaan Oromo nouns in a given text have person, number, gender and possession markers which are concatenated and affixed to a stem or singular noun form.

In addition, Afaan Oromo noun plural markers or forms can have several alternatives. For instance, there are more than ten very common plural markers in Afaan Oromo including: *-oota*, *-oolii*, *-wwaan*, *-lee*, *-an*, *een*, *-eeyyii*, *-oo*, etc.). For example, the Afaan Oromo singular noun *mana* (house) can take the following different plural forms: *manoota* (*mana* + *oota*), *manneen* (*mana* + *een*), *manawwan* (*mana* + *wwan*).

The construction and usages of such alternative affixes and attachments are governed by the morphological and syntactic rules of the language[15]. Afaan Oromo nouns have also a number of different cases and gender suffixes depending on the grammatical level and classification system used to analyze them. Frequent gender markers in Afaan Oromo include *-eessa/-eettii*, *a/-ttii* or *-aa/tuu*.

Example:

Afaan Oromo	Construction	Gender	English
Obboleessa	obbol + eessa	male	brother
Obboleettii	obbol + eettii	female	sister
beekaa	beek + aa	male	knowledgeable

3.4. Questions in Afaan Oromo

All languages have different ways in the use of the word order and question particles. However, question statements are constructed with the help of interrogative words and question marks (to indicate the statement is a question), in every language. English language are used, interrogative articles such as *who*, *what*, *where*, *when*, *why*, *how* to construct a questions.

Similarly, Afaan Oromo interrogative particles help to construct a question sentence. Some of the Afaan Oromo interrogative particles are: “*eessaatti*” (where) “*maaliif*” (why), “*yoom*” (when), “*maali*” (what), “*akkamitti*” (how) and so on. These interrogative particles are used to construct user query.

CHAPTER FOUR

4. MODEL QUERY ANSWERING OVER WEB OF DATA FOR AFAAN OROMO

4.1. Proposed System Architecture Overview

In this chapter, we describe our system in detail and discuss potential implementation issues that may take during our implementation. First, we provide an overview of proposed system architecture. Next, each component of the system and an algorithm to build query execution are discussed in details.

Searching information on the web of data requires user friendly approaches, similar to the ease of tradition keyword-based search engine[60]. In this research, typically Query answering over web of data is proposed to retrieve the best possible answers for end users. The existing approaches of Query Answering (QA) over web of data is designed for non-Afaan Oromo languages [52] [27]. However, none of those efforts were designed to support the Afaan Oromo language which has different morphological and semantics structures. Now days, with the development of the semantic web, a lot of structured data have become available on the web in the form of knowledge base with different format. Making this valuable data accessible and usable for each user query, it is necessary to design query answering over web of data for Afaan Oromo Language. For example Afaan Oromo Wikipedia is considered as part of web of data. Afaan Oromo Wikipedia article contain rich accurate data encoded in millions of table. However, these tables are presented in semi-structured format, they are intended for human consumption are not directly machine readable. The individual facts that each such table encodes are not readily recoverable by automatic methods and cannot be queried by user.

The general architecture proposed for Swoogle [61] and Watson [62] is based on classical semantic web search engine, but there is a limitation to answering a user query. The architecture of the query answering system proposed in this thesis followed the general information retrieval architecture[4]. It can be divided into three main components; the indexing component, the query processing components and the matching components. Unlike the classical search engines, query

answering index the web of data sources which are represented in different data file format like owl file, csv file, rdf file and others.

The indexing component in this case involves integrating these data sources and converting different data representations into a common universal data model. This creates a single coherent knowledge graph that helps to answer queries by combining answers from different data sources to convert each web resources from data source into knowledge graph. The second component is Query processing. The main purpose of this component translating natural language Afaan Oromo queries into a SPARQL query. The last component is answer extraction.

The system process is briefly explained as follows: when the user inputs a query expressed in Afaan Oromo, the query is handled by the query translation module which is the core processing components of Query Answering over web of data, and translates, it to a SPARQL, which is then executed over the RDF knowledge base. The resultant query is executed over the KB to retrieve answers. In the following sections, each components of the system are explained in details as illustrated in figure 4.1.

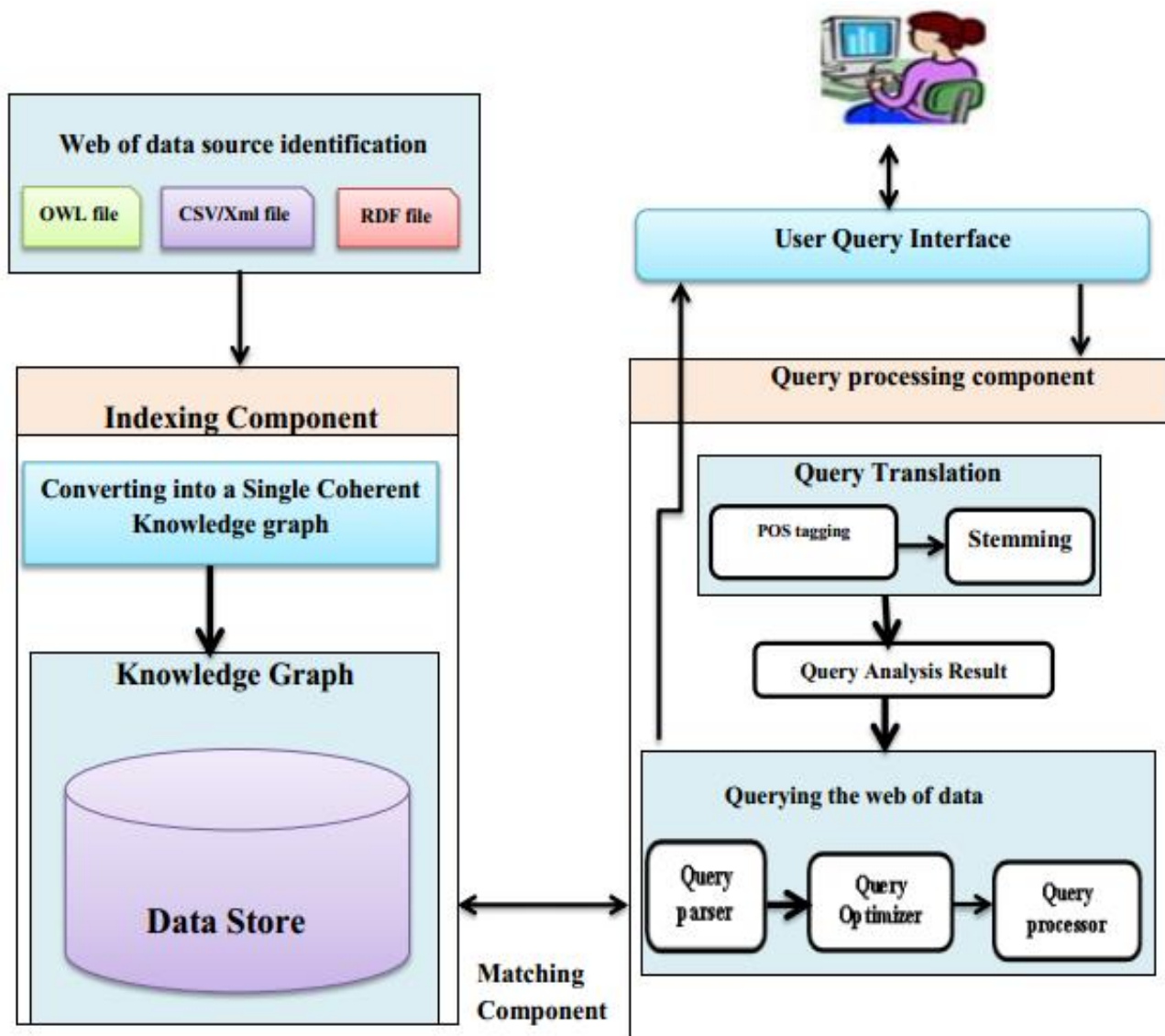


Figure 4.1: Proposed System Architecture

Description of Major Components

In this section we present a details description of each component in our system as follows.

4.1.1. Web of data source identification

With the rapid development of the semantic web project, datasets published using semantic web technologies is increasing more than any time else. Currently, there are the query process on the semantic data is moving forward from one single many-independent data sources distributed across the web. In the web, everything is dominated by data. However, in the present day, data suffers from machine understandable and analyzable. Several attempts have been made to denote data into structured format and resolve their issues. But, many times data is not stored into properly structured format. To making data understandable by both machine and human (i.e. structured format), Software tools that convert from traditional data formats to more reusable formats like RDF are already out there. In this study we have converting the CSV data into RDF using “Ingest tool”. The useful resources are extracted and every member of CSV file is converted into members of domain ontology which means structured format.

There are also semi-structured knowledge sources such as Afaan Oromo Wikipedia that can be easily converted to RDF knowledge graphs. This shows abundance number of Afaan Oromo documents is being published as part of the web of data. Moreover, a query written in Afaan Oromo may also get answer from the web of data documents written in other languages in query answering systems. For instance, a user who wants to visit Bahirdar may be interested to know its temperature. However, he doesn't know how to write his query in Amharic so he writes it in Afaan Oromo by saying “ *Tempiirecharii guyyaa hardhaa Bahirdar meeqa ta'a?* “ which is equivalent to what is the amount of temperature in Bahirdar. The answer to this query is a numerical value so that a person who cannot understand Amharic can also understand the answer. The answer for this query can be extracted from non-Afaan Oromo web of data language sources.

In this section, web of data identification component is responsible to identify and collect different semantic data. The semantic data is stored separately in different data representation models and formats [57]. Recently, semantically rich knowledge bases have become part of the web of data. Examples are Yago[63], Freebase[64], and DBpedia[65]. DBpedia interconnects

hundreds of RDF data source with a total of 30 billion subject-property-object (spo) triples. These data become more tightly interrelated as the number of links in the form of mapping is growing. But, the diversity of data source makes it difficult for user to search and discover relevant information.

4.1.2. Converting into a Single Coherent Knowledge Graph

The major part of the data generated in the web of data have been contributed independently and uses different data representation format[66]. Some of these data are not machine understandable i.e. machine cannot process these data. So some preprocessing tasks are needed to make this data machine understandable, some of them are in the form of traditional data formats such as CSV which make web scale querying as a challenging devour. Therefore, this component of the system will help to create a single data model (i.e. RDF) by converting the existing data represented in heterogonous data formats. It will also create a single coherent knowledge graph among web of data documents distributed across the web.

4.1.3. Knowledge graph

The main idea of the semantic web is to use information from anywhere on the web[39]. When a query is executed, it is infeasible to locate every data source in advance. Therefore, in order to answers from the web of data the first step should be each data source is stored in RDF graph. The task of knowledge graph is to collect as many data sources as possible those may be relevant to answer query. Based on different features of current RDF data source on the web, there are two ways to locate globally distributed source, fist links between different graphs and the second is depending on the semantic web search engine.

We can rely on RDF links to locate other data source. RDF links take the form of RDF triples; in which subject is a URI reference with the namespace one data source and Object is a URI reference with the namespace of other one. When defining a global query, we should simultaneously set a starting point (an RDF graph or a merge of set of graphs) from which we can crawl the web of data to collect more data sources. In addition, graphs may be discovered through the semantic web search engines (e.g. sindice, swoogle etc). They can be used to collect

the graphs those include the same keywords or URIs. In general, knowledge graph components task through search engines will collect the set of statements with the same triple pattern as the query.

One of the limitations of semantic web search engine it uses only structured data. Making this valuable data accessible and usable for end users is one of the main goals of Query Answering over Knowledge bases (KBs). Most current Query Answering systems query in English language. The existing approaches are not designed to be easily adaptable to new KBs and other languages[52].

In our approach, we present new approach for extracting user query from Knowledge base by translating user query to SPARQL queries. The core purpose of KBs is to retrieve the desired information from one or many KBs using natural language questions. This addressed by translating user query to a SPARQL query. The current Afaan Oromo search engine does not address the challenges of query answering over web of data. The reason is that, a lot of Afaan Oromo documents over the web of data have available in unstructured data format. Therefore, it is difficult or impossible to search unstructured data semantically.

4.1.4. Querying the Web of Data

Currently, SPARQL[67][45] is a W3C recommendation and has become the standard language for querying RDF documents. It is used to retrieve data in RDF repositories and has a similar structure as SQL. A RDF data are stored in the form of triples in knowledge base and a SPARQL query is allows for a query to consist of triple patterns.

A general SPARQL query components receives SPARQL queries from the users, process them matching the RDF triples in the knowledge base and then returns the answer to users. A SPARQL query, in its basic format, consists of two parts the SELECT clause identifies the variable to appear in the query results and WHERE clause provides the basic graph pattern to match against the data graph. The WHERE clause consists of one or more triple patterns <S P O> where S, P, O denote the Subject, Predicate and Object respectively.

For example consider the given SP query:

Select ?X

where {?X <foaf:name> ?Y .}

where the resource in the subject is denoted by **X**, the predicate is the resource denoted by the **URI: foaf:name** and the object is the literal value **Y**. The proposed SPARQL query has three subcomponents: Query parser, Query Optimizer and Query Processor. We briefly describe each section as follows:

Query Parser: This subcomponent obtains input queries from users, extracts their BGPs (Basic Graph Pattern) for query optimizer and creates a variable list for the query processing step. In this section we consider basic SPARQL queries with simple clauses.

Query Optimizer: This subcomponent generate an execution plan for the query. Then the query processing is optimized by evaluating the query patterns in an efficient manner. Triple patterns are arranged in an order such that the matching result of a pattern serves as input for the next pattern in the plan. Since the result of each pattern is checked for validity of every processing step, the number of intermediate results is substantially reduced.

Query processor: the tasks of this component is to finding matching points with the query variables, verifying the matching points and then combining them to retrieve the full answer for whole query from knowledge base.

4.1.5. Query Translation

Although the standard approach to extract answer from RDF data sources would be to run a SPARQL query only expert programmers are able to precisely specify their information need using SPARQL. For non-expert SPARQL users, the only option to query this rich data is by keyword search or http protocols where the former cannot retrieve the required answer and the latter is time consuming [27]. Users are also interested to write natural language queries and get answers without seeing the technical challenges of SPARQL.

In general, the entire user queries will be in the form of natural language to search using the traditional search engine applications, but there is no guarantee that user will satisfy with the outcome results. According to the user, querying the databases in natural language is a very easy method to search for any user on desired data but it might be difficult to understand the natural language query for those who are not expert on SPARQL language[68].

With the respect of query translation component, several NLP techniques can be used to convert the user query to SPARQL[56]. In particular, the idea of AUTSPARQL [64] is an interesting solution to convert a natural language query to SPARQL query, which can then retrieve the answers of a question from a given triple store. There are attempts approaches such as GATE: AUTOSPARQL [65] to natural language a user queries into a SPARQL.

An approach is also proposed in QALD-4[69] that support for multiple languages by annotating the question with the set of keywords in an XML or RDF format. Furthermore, most existing research on the area of Knowledge based (KB) query answering (QA) system use the following strategy: (i) analyze the query; (ii) map the entity and predicate from user query to the words in KB; (iii) formulate and select queries; (iv) Search queries and extract answers. However, translating natural language queries into SPARQL query process is not an easy task and it involves language dependent activities starting from the identification of variables in NL query, mapping the variables with the ontology concepts and arranging the mapped variables in a logical SPARQL syntax. Approach mentioned in [70] solve several disambiguation tasks jointly for example: segmentation of query into phrases; mapping of phrases to semantic entities; class and relations and construction of SPARQL triple patterns.

Designing a query translation mechanism for Afaan Oromo language is beyond the scope of this research. However, this research demonstrates the importance of this component. Our strategy of Query Answering for over web of data is not to implementing software from scratch to convert the user query to a SPARQL query; instead we developed manual rule for simple sentence to translate natural query into SPARQL query with the help of, part of speech tagging (POS tagging) and stemming as described below.

Part of speech (POS) tagging is the way to assigning each word in sentences a tag that describes how that word is used in the sentence [71]. POS tagging is necessary to identify verbs, which often represent predicates in RDF triples, and nouns, which often map to ontology classes and instances. At these components we aimed to translating natural language queries into SPARQL queries and extract queries from knowledge base. The method is based on the use of natural language processing (NLP) techniques and resources for enriching queries with linguistic and semantic information. The natural language queries are then translated into SPARQL query with a rule based method approach.

Our method no goes to compare with other methods. Instead, we develop a rule for a natural language queries with the help of POS tagging to assigns a word in the ontology. Part of speech distributions for each word can be derived from the ontology. Since there is no tag set prepared NLP to ontology for Afaan Oromo language, eleven tags have been identified for this study as shown table (4.1)

Table 4-1: Parts of speech (POS) representation

Tags	Description
NN	Noun, singular
NNS	Noun, plural
NNP	Proper noun singular
NNPS	Proper noun, plural
POS	Possessive ending
RB	Adverb
VB	Verb
WP	Wh- pronoun
DT	Determiner
JJ	Adjective
IN	Preposition or subordinating conjunction

Stemming

Stemmer is used for many natural language processing; especially it is very significant for developing, query answering, search engine, machine translation and speech recognizer. At this component, a stemming system for Afaan Oromo is presented. Stemming in Afaan Oromo language is difficult compared to English. English languages have very little inflection and have a tendency to have words that are very close to their respective roots. The destination between the word as a unit of speech and the root as a unit of meaning is more important in the case of language where roots have many different forms when used in actual words as is the case in Afaan Oromo[54]. The goal of stemming is to reduce the number of unique words in the index which in turn reduces the storage space required for the index and speeds up the answering query process and improves the recalling by reducing all forms of words to a base or stemmed form.

In this study, we not used any stemmer proposed for Afaan Oromo, instead we stem all terms and text in Ontological Dictionary to their stem form to compare them with words in user query for tested query. This involves the following steps to extract the root from Afaan Oromo words. For example a stemming for Afaan Oromo should stem the words (barsiisota, barsiisa) to its word stem (barsiis). This comparison makes the mapping much easier and increases searching user query.

4.1.6. User Query Interface

In order to retrieve data from a knowledge base (KB), user query face the challenge of having to learn specific query language (e.g. SQL and SPARQL). However, the rapid growth of query languages to different types of KBs (e.g. Triple Stores) makes it hard for users to keep up with the latest developments of such query languages that allow them to access the data they need for the user. This situation is only possible for SPARQL expert user to access the information from the knowledge base (KB).

Developing user friendly User Query Interface will make it easier for non-experts SPARQL users to access the information in the knowledge base [72].

In this section, we present a User Query Interface that allows users to query the underlying KBs with natural language questions. Our system presents a natural language query interface that takes queries expressed in Afaan Oromo language from the user query and translates it into executable queries which means SPARQL. Finally, returns answers from knowledge base in order to help user to understanding of query.

Constructing Rule for developing an Algorithm

For this thesis to translating natural language query into SPARQL query we construct the rule with the help of linguistic knowledge. The linguistic knowledge plays an important role to construct the rule. The linguistic knowledge required for the natural language query obtained in different ways. One is classifying input query into subject, object and predicate. The rules to develop an algorithm extract from the structure of collected queries in forming meaning of words. The approaches[15] presents the most common way of representing language via rule. The rule underlies many linguistic experts of the language which turn into a set of rules. We have analyzed the structure of user query with respect to SPARQL query structure to develop an algorithm.

Our idea for translating NL queries into formal ones is to specially consider nominal phrases. We observe that a NL query can usually be viewed as the combination of multiple nominal-phrases pairs. From the given NL queries each pair have a noun, verb phrase, a proposition, conjunction etc. At the same time, nominal phrases or words also play an important role in ontologies where store facts to model a domain. The facts are explicitly or implicitly stated in the triple form <subject, predicate, and object>.

The subject and the object may be classes, instances or literal values and usually should be named with nominal words or phrases. The predicate may be prepositions, verbs, verb phrases and so on and sometimes may also be nominal phrases. A nominal-phrases pair represents some kind of semantic relationship between the two nominal phrases. We expect it to be mapped to a triple in the ontology.

Let **C** be the set of all classes, **P** the set of all properties, **I** the set of all instances, **L** the set of all literal contained in the target knowledge base of the SPARQL queries at hand.

We define the transition function:

P: E* \longrightarrow <S P O>* as a function that maps an ontology entity E, or a sequence of entities, to one or more RDF triple pattern(s). For the transition function **P**, the **input** is the sequence Ontology entities recognized by the Query Mapper and the **output** is a set of RDF triple patterns. For this thesis we choose a rule based approach to achieve our objective as follows: In general we categories into two:

a. Rules define how the atomic types (i.e classes, instance, properties and literal) represented in the SPARQL query as follows:

Rule #1: if $x \in (P \cup I \cup L)$ then $P(x) \Rightarrow x$

Rule #2: if $x \in C$ then $P(x) \Rightarrow ?Var \wedge (?Var \text{ rdf:type } x)$

Rule #1: indicates that the properties, instances and literal values remain unchanged in the generated SPARQL body.

Rule #2: indicates that an ontology class entity is represented as a variable **?Var**, that is of type x. the variable name **?Var** is randomly generated.

The procedure of constructing a SPARQL query from Afaan Oromoo natural language query is explained in what follows: For example consider the query “*barsiisota herrega barsiisan baasi*”.

Step1: the query text is mapped into the ontology content. Entities are ordered according to the occurrence of their corresponding words in the query.

Step2: the sequence of ontology entities are scanned for a complete triple pattern. A complete triple pattern <S P O> should fulfill the following conditions.

- It is a sequence of ontology terms that map to <S, P, O > in a sequence.

- A subject can be either a class or an instance
- An object can be a class, an instance or a literal value
- A predicate can be either an Object properties or DataType properties
- The subject and object should belong to the domain and range of the predicate respectively.

If a complete triple pattern is succeed based on the above conditions, the interpretation of the natural language is straight forward into **Rule #1** and **Rule #2** are applied according to the type of each ontology entity, and results are linked together to form one or more triple patterns. The generated triple patterns will formulate the WHERE clause of the SPARQL query. The translation function P of a complete pattern $\langle S P O \rangle$ can be expressed as follows:

Rule #3: $P(S P O) \Rightarrow P(S) \wedge P(P) \wedge P(O)$ where $S \in (C \cup I)$, $p \in P$, $O \in (C \cup I \cup L)$, $S \in$ Domain of P , $O \in$ Range of P . Referring to our example; a subject(*:barsiisota*), a predicate (*:barsiisan*) and an object(*:herrega*) appear in sequence.

b. Rules for translating natural language query into RDF triple as follows:

Rule #1: the first step is to tokenize the user query. Because, the string representing the query is split into tokens which each represent a single word of the query. For example, the query “*barsiisota herrega barsiisan baasi*” yields the following tokens: “*barsiisota, herrega, barsiisan, baasi*”. The order of the tokens is kept, because it is an important piece of information that we use in the next step.

Rule #2: Part of speech tagging of the query. POS tagging is an algorithm that assign a grammatical category (for example noun or verb) to every word of a sentence. For instance, the query “*barsiisota herrega barsiisan baasi*” could be POS-tagged as follows: *barsiisota* (NN), *herrega*(NN), *barsiisan* (VB) and *baasi* (VB). POS-tagging the query enables us to enrich the query to map the RDF triple into ontology. After we tagged our query we assigned to RDF triple format $\langle S P O \rangle$ to search user query from ontology domain as shown in next rule.

Rule #3: If the first term index tagged from query is zero and noun, then the system assigned to the subject.

Rule #4: If the second term index tagged from query are one and noun, then the system assigned to the object.

Rule #5: If a term after the object is verb, then the system check the verb list which mapped to the current input subject and object and finally, assigned to the predicate.

Rule #6: If the tagged query has more than two nouns, then the system check from noun list the mapped subject and object from ontology with the predicate.

Algorithm to build query execution

In order to access the data on Ontology, we needs Ontology understanding query such as SPARQL. However, end users can't understand and type the format of SPARQL query explicitly when they wanted to search the required natural language query. The end users enter the unstructured sentence as input. So, it is needed to extract the triples (i.e. Subjects, Predicates and Objects) from the input query to build the ontology browsing query SPARQL.

However, there are no existing algorithms for Afaan Oromo to extract triple pattern for the input user query. In order to develop an algorithm presented in this work we set a rule and the aim of this algorithm is to translate natural language query to SPARQL query, extract the specific triples from the ontology knowledge base (KB) and finally produces SPARQL queries as output for the end user.

In general the algorithms proposed for this thesis to handle the query answering over web of data are discussed into two major parts as follows:

a. The algorithm to translate the input query into the RDF triple format

Algorithm 4.1: Translate user query into RDF triple

1. **Input:** Natural language query (user query)
 2. **Output:** RDF triples (Subject, Object, and Predicate)
 3. **Function:** Extract user query from knowledge base
 - 3.1. Tokenize the input query
 - 3.2. Tagging the query to assign a grammatical category (NOUN or VERB) to every word of the sentence to enrich the query to map RDF triple into ontology.
 - 3.3. If the first term index = 0 and the first term = NOUN
Then term = SUBJECT
Else
Term= not mapped in ontology
 - 3.4. If the second term index = 1 and the second term = NOUN
Then term = OBJECT
Else
Term= not mapped in ontology
 - 3.5. If the term after object = VERB
Else if term = mapped in ontology
Then term = PREDICATE
Else
Term = not mapped in ontology
 - 3.6. If NOUN greater than two
Then
Term = subject, object, predicate mapped together in ontology
 4. **Result:** RDF triple (Subject, Object, predicate)
-

b. The algorithm to extract RDF triple from knowledge base and display for end user

Algorithm 4.2: RDF Triple-pattern-Extraction

1. **Input:** Natural Language Query
 2. **Output:** Subject, predicate and object
 3. **Function:** Triple Extraction (Query sentence)
 - 3.1. Create list for each word
 - 3.2. Concept list <- Define type-of concept (list) <by ontology>>
 4. Object <- extract –OV (Concept list)
 5. Subject-group <-list
 6. Subject <- Extract-sp (subject-group)
 7. result <- result U subject U object
 8. **Return** result
-

Algorithm 4.3: Extract-SP (Subject -group)

1. **Function:** EXTRACT-SP (subject-group)
 2. While (predicate. Exist)
 3. Find the class of predicates <by ontology>
 4. Cluster the predicates with the same class
 5. Subject <- class name
 6. If subject is equal to class-type (word) in the list
 7. Remove this class-type (word) from the list
 8. Predicate <- cluster of predicates
 9. Remove predicates from the list
 10. Define the relation based on range value <by ontology>
 11. Result <-result U subject U predicate U relation
 12. While (class. exist)
-

13. Subject <-class name
 14. Remove this class-type (word) from the list
 15. Predicate <-string type of data-type properties
 16. Define the relation based on range value <by ontology>
 17. Result <-result U subject U predicate U relation
 18. **Return** result
-

Algorithm 4.4: EXTRACT-OV (Concept List)

1. **Function:** EXTRACT-OV (concept list)
 2. While (constraint. exist)
 3. If consecutive constraints are found then
 4. Decide on those constraint should be combine or not
 5. Object <- constraint. Class <by ontology>
 6. Define relation based on range value <by ontology>
 7. Predicate <-constraint. Data type –properties <by ontology>
 8. If object and predicate are equal to class-type (word) in the list
 9. Remove resulted object and predicates from the list
 10. Result <- result U object U predicate U constraint U relation
 11. **Return** result
-

CHAPTER FIVE

5. EXPERIMENTATION AND EVALUATION

5.1. Overview

Thus in this chapter, the set of experiments conducted to validate the proposed Query Answering over Web of Data approach are presented. We have followed some set of procedure to conduct the experiment.

5.2. Experiment procedure

The following procedures are used for testing the approaches proposed in this thesis.

5.2.1. Prototype Implementation tools

To implement the prototype of “AFAAN OROMO QA OVER WEB OF DATA” we need different components at different phases for retrieving the user query. These components will be used to understand the semantics web behind the query identify the relationship between the query components and transform to SPARQL. For this reason, we used various kinds of open source software tools as follows.

For ontology building, we used *protégé version 5.2.0* which is very popular tool for editing and managing ontologies. *Java Development Kit (JDK) 1.6*: A software development package from Sun Microsystems that implements the basic set of tools needed to write, test and debug Java applications. *Eclipse Standard/SDK version 7.3.0*: This is the program which helps us to build and finish the system implementation using java language. To implement the queries, **Jena framework 2.6.2** has been used. Jena is a free and open source Java Frame work for building semantic web and linked data application. Jena provides an API for creating and manipulating RDF models. It contains also an internal Triple Store, an ontology API and of course SPARQL support[7]. **Ingest tool** to convert CSV file to RDF format. Ingest tool is used to generate RDF file when CSV data are used as an input in the tool and the resulting can be displayed in RDF format. Finally, we have used Stanford NLP for tokenization, POS tagging and stemming.

5.2.2. Experimental Dataset

For the experimental purpose, we have collected some experimental datasets from Afaan Oromo education domain and the data has prepared in two forms as follows: First, Converting non-structured file (e.g. CSV file, xml file etc.) into RDF file format and Second, sample Afaan Oromo ontology domain are developed. Details are discussed in the next section.

Converting non-structured file into RDF file format

In the experimental stage, the CSV file implementation report data sets are used for testing the transformation process and Afaan Oromo education datasets are used for measuring the algorithmic efficiency. Our prepared CSV test data files are saved as shown in table 5.1.

Table 5-1: Sample CSV test data files

Deetaa qindaa'ee/dataset	Lakk-faayila/ No of files	Hanga-deetaa/ Total size(KB)	Lakk-tooraa/ Total no rows
Odeeffannoo Barsiisaa/ Teacher Information	1	3	41
Odeeffannoo baraataa /Student Information	1	3	50
Odeeffannoo gosa-barnoota/ Course Information	1	2	55

Developing Sample Afaan-Oromo Ontology Domain

The proposed Afaan –Oromo query Answering system is generic in terms that it should accept any domain ontology represented in Afaan-Oromo language. Due to the lack of ontologies that are built using Afaan-Oromo language, we built ontology that covers a restricted domain of knowledge from education domain.

We used this ontology throughout this thesis to discuss our proposed approach to convert natural

language queries to SPARQL queries and to demonstrate sample results. However, we emphasize that the system can work with related Afaan-Oromo ontologies domain that follow the same constraints.

There are different tools available for developing ontologies like OilEd, OntoLingua, Apollo, OntoEdit and protégé etc [73]. We have used protégé which is one of the most widely used ontology development editor that defines ontology concepts (classes), properties, taxonomies, various restrictions and class instances. It also support several ontology representation language, including OWL[74]. We have developed the ontology contents for education domain, collected from a number of relevant research papers, Afaan Oromo website and documentations of education domain etc. Ontology is the main term to link web of data. The main purpose of ontology is to capture the domain knowledge in such a way that it is easily understandable by machines. For experiment purpose we select education domain. The ontologies in education system are very emerging these days. In the domain of educational system the ontology delivers the information of how the classes are related to each other.

When it comes to the concept of “Education” then ontology is created on the basis of University. Different group of developers create their own ontology. But if we compare those ontologies some concepts will be always missing because, every developer has different requirements.

There is some ontology related to University in education domain such as developing University ontology. In our work, we have focus on the university employee and ontology based on the course, which focus on particular course to reuse of course for teaching purpose.

The steps to building ontologies for education domain as shown below

Step1: Determine the domain and scope of the ontology

Step2: Defining overview of the ontology

Step3: Defining classes and class hierarchy of ontology

Step4: Defining the properties of classes (slots)

Step 5: Creating instances/Individuals

Each step is discussed as follows:

Step1: Determine the domain and scope of the ontology

The first steps of ontology development consider, in which the ontology will be developed in order to answer some basic questions:

1. What is the domain that the ontology will cover?

The domain of the ontology will cover study of education in general

2. What is the use of the ontology?

The ontology is to provide a schema-base of education, which is used in our query answering system to retrieve the exact answer to the user query.

4 What types of questions would be answered by the information contained in the ontology?

The ontology would provide answers to questions relating to education domain for Afaan Oromo, like:

- Barsiisota herrega barsiisan baasi/ list teachers who teaches maths
- Baraattoo herrega baraataa baasi/ list studentes who learn maths

Step2: Defining overview of the ontology

We designed our ontology from scratch as a prototype model to serve in implementation of our system. We identify education data that are needed in the process of query answering in our system. Our ontology was represented in OWL and contains general characteristics. It can adds or update any information about the education easily and without affecting the overall structure

of the ontology whether at the level of classes or properties. Table 5.2 shows number of classes, object, properties and data properties in our ontology.

Table 5-2: Classes, Object Properties and Data properties in the domain

Domain and scope of the ontology	Education
Number of classes	5
Number of object properties	6
Number of data properties	7

Step3: Define classes and class hierarchy of ontology

This sections defines classes (concepts) used in our ontology domain. These classes are not selected randomly, but they are selected depending on our domain “Education”. Table 5.3 shows the ontology classes.

Table 5-3: Ontology classes

No.	Class/Afaan – Oromo	Class/ English	Description
1	Yuunivarsitii	University	University is subclass of thing and super class of other classes including course (gosa barnoota) and people (nama). The university will contain information about these classes relevant to the university domain.

2	Nama	People	The class of person contains all the personals and the students emerged with the institution in any ways. This class will have all information about the people for example age, date of birth, personal number, student number, teacher name etc.
3	Gosa-barnoota	Course	Course is the subclass of university. In this class all the courses which are offered by the institution in the specific year are mentioned. The course will have course name and course code. The course will also have information about the year in which the course has been offered, the professor who will be teaching the specific course in specific year and information about the students taking the course.
4	Barsiisaa	Teacher	This class contains all the information about the teachers. This class will also have information about the regarding department of teacher and the course which will be responsibility of teacher. These classes have subclasses which are professor and assistant professor.
5	Baraataa	Student	In this class student of all levels will be mentioned alongside the information about their student ID, programme registered in, and department etc. the student is further divided into subclass as follows: first degree (diigrii jalqaba) and second degree (diigii lammaattaa).

Table 5.3 Contains five ontology concepts mentioned in our domain “education”. Choosing these concepts has direct relation with user requirements used in the process of query answering in education domain. We mention every ontology domain in Afaan-Oromo including the description of the concepts. Figure 5.1 shows the ontology classes hierarchy for education domain in protégé.

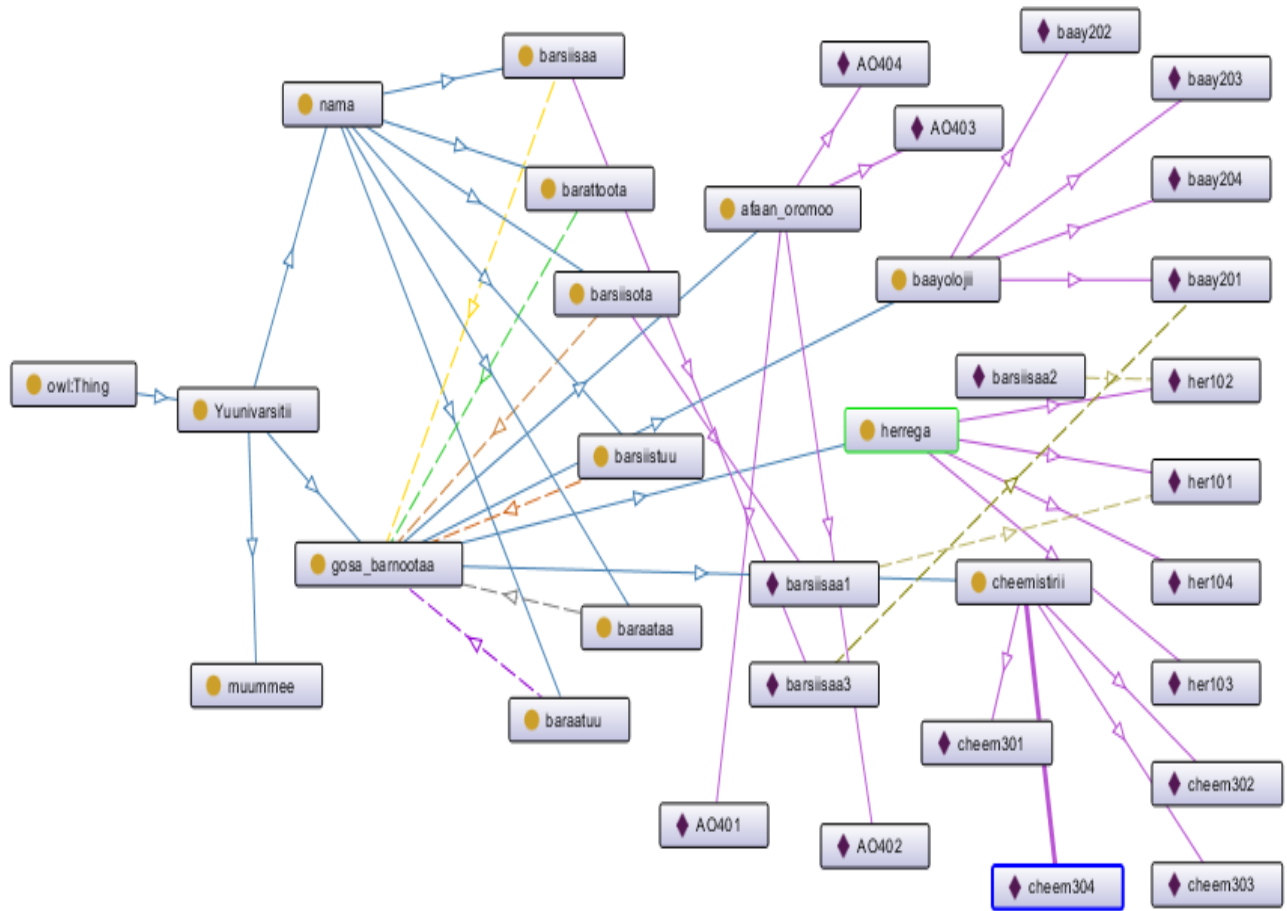


Figure 5.1: Class Hierarchy using Protégé for education domain.

Step4: Define the properties of classes (slots)

The other requirement of creating ontology is object properties (relation) and data type properties. Creating object properties plays a vital role in connecting classes (concepts) of the ontology in our education domain. We used 5 object properties that connect the important concepts which have relations with each other that are illustrated in table 5.3.

Table 5-4: Object properties in Education ontology

No.	Object properties/ Afaan –Oromoo	Object properties/ English	Domain	Range
1	Barsiisan	Teaches	Nama/people	Gos-barnoota/course
2	Baratu	Learn	Nama/people	Gos-barnoota/ course
3	Barsiistu/female	Teaches	Nama/people	Gos-barnoota/ course
4	Barsiisu/male	Teaches	Nama/people	Gos-barnoota/ course
5	Galmahe	Register	Nama/people	Gos-barnoota/ course

Table 5-5: Data type properties in Education ontology

No.	Datatypeproperty/ Afaan-Oromo	Data Type property/English	Domain	Range
1	Lakk-baraataa	Student number/male	Baraataa	XSD:integer
2	Lakk-baraatuu/	Student number/female	Baraatuu	XSD:integer
3	Lakk-barattoota	Students number	Barattoota	XSD:integer
4	Lakk-barsiisa	Teacher number	Barsiisaa	XSD:integer
5	Lakk-barsiisota	Teachers number	Barsiisota	XSD:integer
6	Maqaa-jalqabaa	First name	Nama	XSD:string
7	Maqaa-Abbaa	Last name	Nama	XSD: string

Step 6: Creating instances/Individuals

The last step is creating instances (individuals) of classes in the hierarchy. The creation of individuals allows all the properties of the classes to record. Instances are created and stored in a separate database named RDF store in format of RDF triples. Our system is designed to keep the data separate from the ontology which should remain intact and reusable. The information of individuals is taken from documentations of education domain. In our ontology, we defined around 347 instances representing all ontology concepts.

Ontology Testing Question

For testing purpose, we could not find any Afaan Oromo test data that is used for Ontology based query answering in Afaan Oromo. Therefore, we had to build our own test data. We have developed the ontology contents for education Domain. The Ontology is implemented with protégé. In OWL format and can be mapped into acknowledge base. We develop manual ontology for Afaan Oromo to validate the contents. In our ontology we created and stored instances in a separate format of RDF triples.

We performed a series experiments to demonstrate sample results. All our experiments depend on the data in ontology domain and individuals that created system development. We have collected 50 queries about education domain from different source and we have tested 20 queries for experimentation purpose. Then we incorporated these questions into our system and tested them for answers. Table 5.6 shows Sample query tested for experimentation.

Table 5-6: Sample query tested for experimentation

No. Query	Query/Afaan-Oromo	Query/English
Q1:	Barsiisota herrega barsiisan baasi	List of teachers who teaches maths
Q2:	Baraattoota herrega baratan baasi	List of students who learn maths
Q3:	Baraataa afaan oromoo baratu baasi	List of student who learn maths
Q4:	Baraataa herrega galmahe baasi	List of student how register maths
Q5:	Barsiisaa herrega barsiisu baasi	List of teacher who teaches maths
Q6:	Barsiisota waggaa-tokkooffaa barsiisan baasi	List teachers who teaches first year
Q7:	Barattoota waggaa-tokkooffaa hunda baasi	List all first year students
Q8:	Barsiisa diigrii-jalqaba qabu baasi	List teacher how have first digri
Q9:	Barsiistu waggaa-laammaaffaa barsiistu baasi	List teacher who teaches second year
Q10:	Baraattuu herrega galmoofta baasi	List of student how register maths
Q11:	Baraattu Ingiliffaa baratu baasi	List students learn English(Female)
Q12:	Barsiistu Ingiliffaa barsiistu baasi	List teacher teaches English(female)
Q13:	Barattoota Afaan-Oromo qoratan baasi	List students studies Afan Oromo
Q14:	Barsiisota Afaan-Oromo qoratan baasi	List teacher studies Afan-Oromo
Q15:	Baraattuu Afaan-Oromo qoraattu baasi	List student studies Afan-Oromo(female)
Q16:	Baraataa herrega qoraatu baasi	List student studies maths (male)
Q17:	Barsiisa keemistii qoraatu baasi	List teacher studies chemistry (male)
Q18:	Barsiistu keemistii qoraattu baasi	List teacher studies chemistry (female)
Q19:	Barsiisa Afaan-Oromo galmahe baasi	List teacher register Afan-Oromo
Q20:	Baraattu waggaa-sadaffaa baratu baasi	List teacher who learn third year(female)

In general, we tested our system with 20 different natural language queries as shown in table 5.6. These 20 queries were carefully chosen to test the different routes of the algorithm used to

translate natural language queries to SPARQL.

5.3. Prototype

This section will give a description of the implemented prototype. In section (5.2.1) development tools has been used for the implementation is described. Then, after generating the code OWL, We have undertaken the development of a prototype called “**AFAAN OROMOO QUERY ANSWERING OVER WEB OF DATA**” we pass to its operational in JAVA program, used the jena API that serves as a link between OWL code and a JAVA program. In the following we will present some screen captural of our application. Once the application is launched the following interface will be displayed. Figure 5.2 shows the screen shoot of the prototype



Figure 5.2: Afaan Oromo Query answering over web of data prototype

The implementation section is then followed by describing the process after searching the user query answering the ontology is displayed in triple format. For example, consider the user query “barsiisota herrega barsiisan baasi” which means “list teacher teaches mathematics course” where (barsiisota=**subject**, herrega = **object**, barsiisan= **predicate**) finally the system searches from the ontology and display the query to the user as shown in figure (5.3).

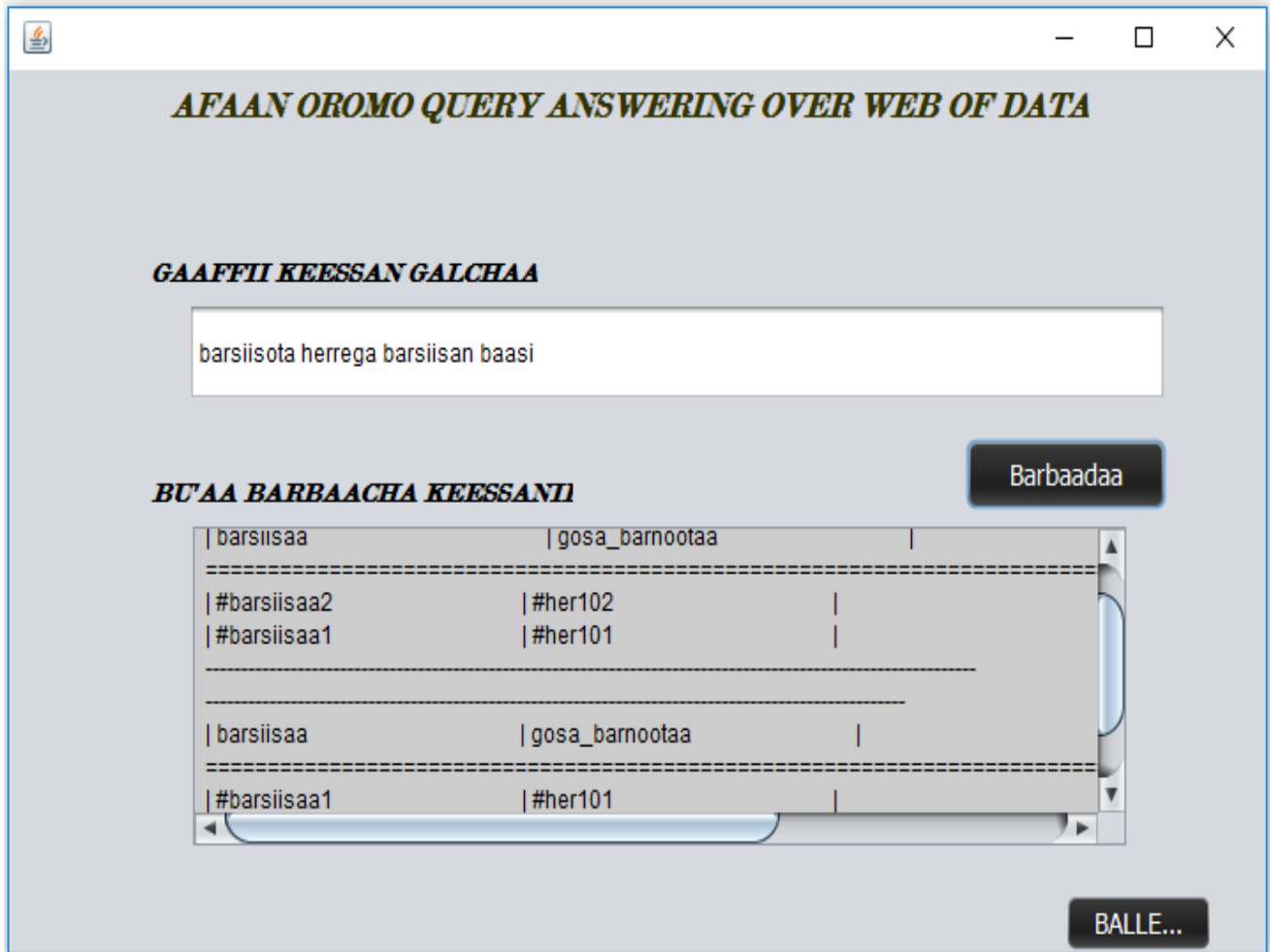


Figure 5.3: Afaan Oromo Query Answering over web of data display the result

Figure 5.4 check the validation of the system. For example, consider the user query “*barsiisota herrega baratan baasi*” which means “*list teacher learn mathematics course*” This types of query is executed when the user insert unmatched query into the system.



Figure 5.4: User interface for checking Validation

5.4. Evaluation parameters

This subsection describes the evaluation parameters of our system to determine whether it behaves exactly as we expect. The main goal of this evaluation is to check the system's ability to translate natural language queries to valid SPARQL queries and retrieve answers from the RDF knowledge base for Afaan Oromo language.

We have used the most common evaluation parameters such as Recall, Precision and F-Measure. Recall is the ratio of the number of relevant answer for query Q to the total number of relevant answers for Query Q whereas precision is the ratio of the number of correct retrieved answers for query Q to the number of relevant answers for query Q. F-measure is the standard measure for evaluating IR by combining recall and precision techniques

The evaluation depends on rules and procedures in our approach of translate the Afaan Oromo queries into SPARQL queries and finding correct answers for queries related to the Ontology domain. The system evaluated with respect to precision and recall. For each query Q, recall, precision and F-measure was computed as equation follows.

$$Recall(Q) = \frac{\text{number of relevant answers for } Q}{\text{total number of relevant answers for } Q} \quad (5.1)$$

$$Precision(Q) = \frac{\text{number of correct retrieved answers for } Q}{\text{number of relevant answers for } Q} \quad (5.2)$$

$$F - Measure(Q) = \frac{2 * \text{precision}(Q) \times \text{Recall}(Q)}{\text{precision}(Q) + \text{Recall}(Q)} \quad (5.3)$$

On the basis of those measures, overall precision and recall values as well as an overall F-measure value were computed as the average meaning of the precision, recall and F-measures values for all tested queries.

We have tested 20 natural language queries in our related Afaan Oromo ontology domain “Education”. We verify that the goal has been performed according to the rule and procedures that we have set in our work. Table 5.7 shows the number of queries and their evaluation parameters results.

Table 5-7: The number of query and their evaluation parameters

Query number	Evaluation parameters		
	Recall	Precision	F-measure
Q1	0.532	0.8	0.639
Q2	0.553	0.692	0.615
Q3	0.617	0.759	0.681
Q4	0.596	0.893	0.715
Q5	0.702	0.758	0.729
Q6	0.617	0.759	0.681
Q7	0.723	0.735	0.729
Q8	0.656	0.645	0.650
Q9	0.553	0.692	0.615
Q10	0.745	0.857	0.797
Q11	0.702	0.758	0.729
Q12	0.656	0.581	0.859
Q13	0.809	0.895	0.850
Q14	0.532	0.68	0.597

Q15	0.574	0.815	0.674
Q16	0.745	0.8	0.772
Q17	0.702	0.848	0.768
Q18	0.681	0.719	0.699
Q19	0.745	0.714	0.729
Q20	0.766	0.833	0.798
Average	0.6603	0.7617	0.7163
Percentage	66.03 %	76.17 %	71.63 %

5.5. Discussion of the result

The proposed system had been tested with a sample Ontology and a testing set consisting of 20 different simple natural language queries for Afaan Oromo. As for comparison and contrast with other methods we are not aware of any previous works that use ontologies for Afaan Oromo Query Answering over Web of Data system. Because of this, we cannot compare or contrast our methodologies with other researches.

As it is shown in table 5.6 the average values for all the three techniques for proposed system had been discussed. When we put the average values in percentage, the proposed system has 66.03% Recall, 76.17% precision and 72.63% F-measure as illustrated Recall, Precision and F-Measure graph in figure 5.4.

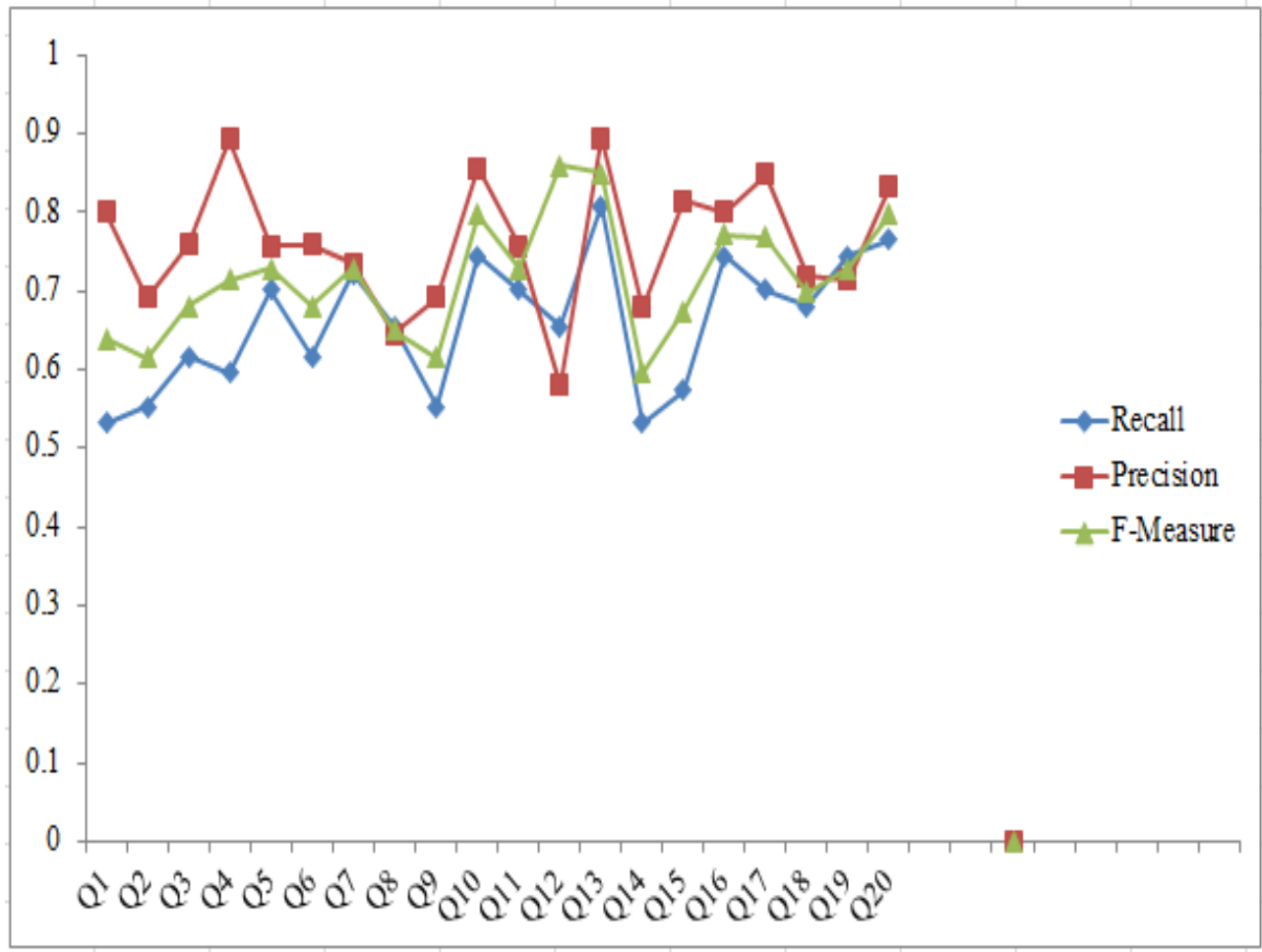


Figure 5.5: Recall, Precision and F-Measure graph

The performance of the system is lowered for various reasons as it is identified by this study. In this study the systems only answer 76.17% precision retrieved queries for those given 20 natural language queries. This is because we have been test only for simple queries. In addition to this, the nature of morphological and semantic structures of Afaan Oromo is very complex. Afaan Oromo terms are highly inflected for gender, numbers, possession, plural and conjunctions.

The above result strengths and limitation we can summarize as follows:

- The system has the ability to translate simple Afaan Oromo NL queries to SPARQL queries

and to identify the RDF triples using the grammar rule we defined.

- The system depends on the ontology knowledge to translate the query and it makes less use of NLP techniques.
- However, the lack of NLP processing may result incorrect translations such as the lack of co-reference resolution which is important to identify expressions and pronouns that map to single entity.
- While the lack of NLP-based can be seen as a limitation, we can think that it is also an important strength point considering the limited support for Afaan Oromo NLP techniques.
- In addition to this, the ability of the system to provide answers depends on the coverage of the ontology and the association knowledge base.
- The more complete the ontology in representing the domain of knowledge, the more the system becomes capable of providing correct answers. Therefore, we think that the inability to answers NL queries due to limited ontology cannot be considered as a limitation in the system.

CHAPTER SIX

6. CONCLUSION AND FUTURE WORK

6.1. Overview

In this chapter, we present the conclusions and future work or recommendation related to the research questions addressed by this study in section 1.4 and discussed as follows:

6.2. Conclusion

This study proposed and designed as system called “Query Answering over Web of Data for Afaan Oromo” based on ontology that translates natural language queries in Afaan Oromo to SPARQL queries and answer user query from web of data.

In our approach, we have built the sample Afaan Oromo ontology model from scratch that handles the natural language queries and translates into triple patterns to create the SPARQL queries and then execute them through Jena framework. The ontology was created using protégé tool which comprises more than 447 instances.

The general architecture proposed query answering system is discussed in three main components; the indexing component, the query processing component and the matching component. The first, component is used to integrate data sources and converting different data representation into a common universal model. The second component takes natural language query as input and translates into SPARQL query. The last component is extract answer from knowledge graph.

Furthermore, we present how to convert non-structured (csv/xml file) query into structured query (i.e rdf or owl file) formats. Also, how to translate the queries into triple patterns and build the SPRQL queries which mechanism to retrieve the answer from knowledge base. In addition, NLP functions have been used mainly POS-tagging and stemming to handle query processing.

This study is the first for Afaan Oromo language on the area of query answering over Web of Data. The system evaluation showed a promising performance being tested on a 20 natural

language query based on ontology for education domain. It obtained 66.03% Recall, 76.17% Precision and 71.63% F-Measure for tested query. Even though, the performance of the system is lowed, this due to the tested query depends only for simple sentence natural language query. From this result, we conclude that developing completeness ontology for natural language query play a vital role the features.

6.3. Future works

Query Answering over Web of data is a new research area for Afaan Oromo language and also for other Ethiopian language. The task is very complex for such under resources language. Important research topics not fully addressed in this thesis or in close relation to the ones we have addressed. It needs collaboration of researcher. The reason is, answering natural language query over web of data vital components to search information from web of data. In this section we discuss unsolved limitations further incremental improvements, as well as interesting research topic that can be perused to enhance the proposed work.

- We look to increase the scope of this research to include more types of query such a comparative and similarity phrases.
- We look to extend the ontology by adding more data and semantic information
- We look to forward to increasing our model and procedures of NL query to cover the largest amount of query and obtain more accurate results.
- In order to automatically create RDF-triples and computational queries for some of those questions, deeper linguistic analysis is required. It is possible to explore and generalize this linguistic analysis in the future.

References

- [1] A. Albarghothi, F. Khater, and K. Shaalan, “ScienceDirect ScienceDirect Arabic Question Answering Using Ontology,” *Procedia Comput. Sci.*, vol. 117, pp. 183–191, 2017.
- [2] H. Yue, S. Member, L. Guo, and R. Li, “DataClouds : Enabling Community-Based Data-Centric Services Over the Internet of Things,” no. October, 2014.
- [3] J. Pan, S. Paul, and R. Jain, “D EPLOYMENT P ERSPECTIVES A Survey of the Research on Future Internet Architectures,” no. July, pp. 26–36, 2011.
- [4] Deo, Arpit, Jayesh Gangrade, and Shweta Gangrade. "A SURVEY PAPER ON INFORMATION RETRIEVAL SYSTEM." *International Journal of Advanced Research in Computer Science* 9.1 (2018).
- [5] B. T. Berners-lee and J. Hendler, “The Semantic Web,” vol. 21, 2002.
- [6] S. Ismail and T. Shaikh, “A L ITERATURE R EVIEW ON S EMANTIC W EB – U NDERSTANDING T HE P IONEERS ’,” no. September, pp. 14–28, 2016.
- [7] M. Sander, U. Waltinger, M. Roshchin, and T. Runkler, “Ontology-Based Translation of Natural Language Queries to SPARQL,” pp. 42–48, 2014.
- [8] M. Richardson and P. Domingos, “Building Large Knowledge Bases by Mass Collaboration,” 2003.
- [9] I. Alagha and G. Strip, “U SING L INGUISTIC A NALYSIS TO T RANSLATE A RABIC N ATURAL L ANGUAGE Q UERIES TO.”
- [10] A. Bouziane, D. Bouchiha, N. Doumi, and M. Malki, “Question Answering Systems : Survey and Trends,” *Procedia - Procedia Comput. Sci.*, vol. 73, no. Awict, pp. 366–375, 2015.
- [11] M. Yahya, K. Berberich, and S. Elbassuoni, “Natural Language Questions for the Web of

- Data,” no. July, pp. 379–390, 2012.
- [12] Tesfaye Guta Debela " AFAAN OROMO SEARCH ENGINE " ADVISOR : Dida Midekso (PhD),” no. November, 2010.
- [13] Berhanu Anbase Bedada, “*Developing Latent Semantic Indexing based Information Retrieval for Afaan Oromo* " Jimma Institute of Technology School of Graduate Studies Department of information Technology,” 2017.
- [14] A. T. Nedjo, “Challenges of Diacritical Marker or Hudhaa Character in Tokenization of Oromo Text,” vol. 9, no. 7, pp. 1818–1826, 2014.
- [15] W. Tesema, D. Tesfaye, and T. Kibebew, “iMedPub Journals Designing a Rule Based Disambiguator for Afan Oromo Words Keywords : Introduction Overview of Afan Oromo Language Afan Oromo Word Senses,” pp. 3–6, 2017.
- [16] K. M. Jimalo, “Afaan Oromo News Text Categorization using Decision Tree Classifier and Support Vector Machine : A Machine Learning Approach,” vol. 47, no. 1, pp. 1–12, 2017.
- [17] T. Tolessa, “Early History of Written Oromo Language up to 1900,” vol. 7522, no. June, pp. 76–80, 2012.
- [18] D. Agrawal, H. Sanghani, S. Jadhav-, and S. Shinde, “Ontology based Domain Specific Web Search Engine,” no. 4, pp. 12–15, 2015.
- [19] A. Halevy, “Answering Structured Queries on Unstructured Data.”
- [20] S. Y. R. Esearch, “Design Science in Information Systems Research,” no. September 2014, 2004.
- [21] K. Peffers, “A Design Science Research Methodology for Information Systems Research,” vol. 24, no. 3, pp. 45–78.

- [22] J. Sj, “Design Science in the Field : Practice Design,” vol. 2, no. 1, pp. 67–81, 2018.
- [23] M. T. Mullarkey and A. R. Hevner, “An elaborated action design research process model An elaborated action design research process model,” *Eur. J. Inf. Syst.*, vol. 9344, pp. 1–15, 2018.
- [24] S. T. March and G. F. Smith, “Design and natural science research on information technology,” vol. 15, pp. 251–266, 1995.
- [25] R. Baskerville, A. Baiyere, S. Gregor, A. Hevner, and M. Rossi, “Design Science Research Contributions : Finding a Balance between Artifact and Theory,” vol. 19, 2018.
- [26] F. Getahun and G. Asefa, “Towards Amharic Semantic Search Engine,” 2015.
- [27] I. Alagha, “AR2SPARQL : An Arabic Natural Language Interface for the Semantic Web,” vol. 125, no. 6, pp. 19–27, 2015.
- [28] S. Brin and L. Page, “Reprint of: The anatomy of a large-scale hypertextual web search engine,” *Comput. Networks*, vol. 56, no. 18, pp. 3825–3833, 2012.
- [29] A. McCallum, K. Nigam, J. Rennie, and K. Seymore, “A Machine Learning Approach to Building Domain-Specific Search Engines,” *Int. Jt. Conf. Artif. Intell.*, vol. 16, pp. 662–667, 1999.
- [30] D. Hawking, “Challenges in Enterprise Search,” *ADC '04 Proc. 15th Australas. database Conf. - Vol. 27*, vol. 27, pp. 15–24, 2004.
- [31] C. Wu, K. Jenab, S. Khoury, and S. Moslehpour, “Journal of Project Management,” vol. 3, pp. 89–104, 2018.
- [32] C. Science, “Ontology based Semantic Search Engine,” vol. 2, no. 8, pp. 1349–1353, 2012.
- [33] L. Nadia, “Design and implementation of information retrieval,” 2014.

- [34] A. Sayed and A. Al, “IBRI-CASONTO : Ontology-based semantic search engine,” *Egypt. Informatics J.*, vol. 18, no. 3, pp. 181–192, 2017.
- [35] M. Chen, D. Ebert, H. Hagen, R. Laramée, W. Ribarsky, and D. Silver, “Data , Information and Knowledge in Visualization.”
- [36] T. R. Gruber and T. R. Gruber, “A Translation Approach to Portable Ontology Specifications by A Translation Approach to Portable Ontology Specifications,” vol. 5, no. September 1992, pp. 199–220, 1993.
- [37] N. Guarino, “Formal Ontology and Information Systems,” no. June, pp. 3–15, 1998.
- [38] A. J. Pretorius, “Ontologies - Introduction and Overview,” pp. 1–13, 2004.
- [39] B. M. C. Daconta, L. J. Obrst, and K. T. Smith, “The Semantic Web : A Guide to the Future of XML , Web Services , and Knowledge Management,” 2003.
- [40] G. O. Klein and B. Smith, “Concept Systems and Ontologies: Recommendations for Basic Terminology,” *Trans. Japanese Soc. Artif. Intell.*, vol. 25, pp. 433–441, 2010.
- [41] T. Tran, P. Cimiano, S. Rudolph, and R. Studer, “Ontology-based Interpretation of Keywords for Semantic Search,” *Iswc 2008*, pp. 523–536, 2008.
- [42] G. Zenz, X. Zhou, E. Minack, W. Siberski, and W. Nejdl, “From keywords to semantic queries-Incremental query construction on the semantic web,” *J. Web Semant.*, vol. 7, no. 3, pp. 166–176, 2009.
- [43] M. Fernández, I. Cantador, V. López, D. Vallet, P. Castells, and E. Motta, “Semantically enhanced Information Retrieval: An ontology-based approach,” *J. Web Semant.*, vol. 9, no. 4, pp. 434–452, 2011.
- [44] C. Rocha, D. Schwabe, and M. P. Aragao, “A hybrid approach for searching in the semantic web,” *Proc. 13th Conf. World Wide Web - WWW '04*, p. 374, 2004.

- [45] M. Arenas and J. Pérez, “Querying Semantic Web Data with SPARQL,” 2011.
- [46] L. Roffia, P. Azzoni, C. Aguzzi, and T. S. Cinotti, “Dynamic Linked Data : A SPARQL Event Processing Architecture,” vol. 0, pp. 1–33.
- [47] I. Journal, O. F. Engineering, A. Ontology, E. Tools, and F. O. R. Effective, “International journal of engineering sciences & research technology analyzing ontology editing tools for effective semantic information retrieval,” vol. 6, no. 5, pp. 40–47, 2017.
- [48] B. Kapoor and S. Sharma, “A Comparative Study Ontology Building Tools for Semantic Web Applications,” vol. 1, no. July, pp. 1–13, 2010.
- [49] M. Palmer, “VERB SEMANTICS AND LEXICAL Zhibiao W u,” pp. 133–138.
- [50] S. R. B. S, “Ontology based Semantic Search Engine for Cancer,” vol. 95, no. 5, pp. 39–43, 2014.
- [51] A. A. L. Muqrishi, A. Sayed, and M. Kayed, “CASENG : ARABIC SEMANTIC SEARCH ENGINE,” vol. 75, no. 2, pp. 148–159, 2015.
- [52] D. Diefenbach, A. Both, K. Singh, and P. Maret, “Towards a Question Answering System over the Semantic Web,” vol. 0, no. 0, pp. 1–15, 1900.
- [53] W. Tesema, “American Journal of Computer Investigating Afan Oromo Language Structure and Developing Effective File Editing Tool as Plug-in into Ms Word to Support Text Entry and Input Methods.”
- [54] D. Tesfaye, “Designing a Rule Based Stemmer for Afaan Oromo Text,” no. 1, pp. 1–11.
- [55] T. Girma, “HUMAN LANGUAGE TECHNOLOGIES AND AFFAN OROMO ISSN : 2278-6252 I . INTRODUCTION,” vol. 3, no. 5, pp. 1–13.
- [56] I. Bedane, “The Origin of Afaan Oromo : Mother Language,” vol. 15, no. 12, 2015.
- [57] Abdi S, "AFAAN OROMO NAMED ENTITY RECOGNITION USING HYBRID

- APPROACH " " Addis Ababa University March, 2015.
- [58] Eshetu Gusare "*Sentiment Analysis for Opinionated Afaan Oromoo Texts*" Addis Ababa University, Ethiopia October, 2017.
- [59] Tesfa Kebede Hundesa "WORD SENSE DISAMBIGUATION FOR AFAAN OROMO LANGUAGE" Addis Ababa, Ethiopia November, 2013.
- [60] E. Rajabi, C. Debruyne, and D. O. Sullivan, "Towards a Personalized Query Answering Framework on the Web of Data," pp. 2–6, 2017.
- [61] T. Finin, R. S. Cost, and J. Sachs, "Swoogle : A Semantic Web Search and Metadata Engine *."
- [62] J. Euzenat, I. G. Rhône-alpes, W. Hall, and M. Keynes, "Watson , more than a Semantic Web search engine," vol. 0, no. 0, pp. 1–9, 1900.
- [63] F. M. Suchanek and G. Weikum, "YAGO : A Core of Semantic Knowledge Unifying WordNet and Wikipedia," pp. 697–706, 2007.
- [64] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase : A Collaboratively Created Graph Database For Structuring Human Knowledge," pp. 1247–1249, 2008.
- [65] C. Bizer, G. Kobilarov, J. Lehmann, and Z. Ives, "DBpedia : A Nucleus for a Web of Open Data."
- [66] A. Kulkarni, M. Sagavkar, A. Elumalai, and N. C. S. N. Iyengar, "Evaluating SPARQL Query on Semantic Data Store in Cloud Platform," vol. 8, no. 4, pp. 251–264, 2015.
- [67] M. D. Nguyen, M. S. Lee, S. Oh, and G. C. Fox, "SPARQL Query Optimization for Structural Indexed RDF Data."
- [68] S. Shaik, P. Kanakam, S. M. Hussain, and D. Suryanarayana, "Transforming Natural

- Language Query to SPARQL for Semantic Information Retrieval,” no. 7, pp. 347–350, 2016.
- [69] C. Unger *et al.*, “Question Answering over Linked Data (QALD-4) To cite this version : HAL Id : hal-01086472 Question Answering over Linked Data (QALD-4),” 2014.
- [70] M. Yahya, K. Berberich, S. Elbassuoni, M. Ramanath, V. Tresp, and G. Weikum, “Deep Answers for Naturally Asked Questions on the Web of Data,” pp. 445–448, 2012.
- [71] G. M. Wegari, “Parts of Speech Tagging for Afaan Oromo,” pp. 1–5.
- [72] D. Song and C. Brew, “Natural Language Question Answering and Analytics for Diverse and Interlinked Datasets,” pp. 101–105, 2015.
- [73] S. Youn, D. Mcleod, and D. Mcleod, “Ontology Development Tools for Ontology- Based Knowledge Management Ontology Development Tools for,” 2006.
- [74] V. Jain and M. Singh, “Ontology Development and Query Retrieval using Proté gé Tool,” no. June 2016, 2013.