Addis Ababa University

College of Natural Sciences

Afaan Oromo List, Definition and Description Question Answering System

Chaltu Fita Elanso

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

April 14, 2016

Addis Ababa University

College of Natural Sciences

Chaltu Fita Elanso

Advisor: Dida Midekso (PhD.)

This is to certify that the thesis prepared by Chaltu Fita, titled*: Afaan Oromo List, Definition and Description Question Answering System* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science compiles with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	Name	Signature	Date
Advisor:			
Examiner:			
Examiner:			

Abstract

Information is very important in our day to day activity. Technology plays an important role in order to satisfy human beings information need through the use of Internet where people ask questions and a system provides an answer for their query. For instance, search engines a user submit a query and the search engine displays a link to relevant web pages for each issued users query. The QA systems emerge as best solution to get the required information to the user with the help of information extraction techniques.

QAS has been developed for English, Amharic, Afaan Oromo and other languages. The Afaan Oromo QAS is developed for answering factoid type questions where the answer is named entity. In this thesis, QAS is developed for answering list, definition and description question which deals with more complex information need. Document preprocessing, question analysis, document selection and answer extraction are the components used for developing the QAS. Tokenization, case normalization, short word expansion, stop word removal, stemming, lemmatization and indexing are the tasks of pre-processing. Question classification is done using a rule based approach. The subcomponents in document selection are document retrieval used for retrieving relevant documents and document analysis used for filtering the retrieved documents. The answer extraction component have sentence tokenizer for tokenizing sentences retrieved from the document analysis and independent subcomponents for definition-description and list were used, DDAE contains sentence extractor for extracting sentences from sentence tokenizer, the answer selection algorithm selects top 6 sentences from the scored and ranked sentences and finally sentence ordering algorithm order the sentences. The LAE contain candidate answer extraction for extracting through rules and gazetters and select answer.

The system is tested using evaluation metrics. We used percentage ratio for evaluating question classification which classified 98% correctly. The performance of document selection and answer extraction is tested using precision, recall and F- score. Document selection component is tested and scored an F-score of 0.767. Finally, the answer extraction component is evaluated with an average F-score of 0.653.

Keywords: Afaan Oromo List, Definitional and Descriptional Question Answering, Rule Based Question Classification, Document Filtering, Sentence Extraction, Answer Selection

Dedication

To the Almighty GOD and my Family

Acknowledgment

All the praises and thanks be to God, St.Mary and St. Gabriel for being with me all the time.

It would not have been possible to write this thesis without the help and support of the kind people around me. First and foremost, I would like to take this opportunity to express my deepest gratitude toward my wonderful advisor, Dr. Dida Midekso, for the advice and guidance from the early stage of this research. I am also thankful for all the support, encouragement and invaluable comments provided to me. I would like to express my heartfelt thanks to Aberash, Tilahun and Michael Gasser for your constructive and professional advices. I also like to thank Mr. Girma Dechassa Ambo University Afaan Oromo department head and staffs for giving me all the necessary information regarding the language.

Finally, I would express my special thanks to all of my family members: my father, mother sisters and brother and my friends for giving me support and encouragements.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Acronyms

AOLDDQAS:	Afaan Oromo List, Definition, Description, Question Answering System
API:	Application Programming Interface
AVG:	Average
DDAE:	Definition-Description Answer Extraction
IE:	Information Extraction
IR:	Information Retrieval
LAE:	List Answer Extraction
NER:	Named Entity Recognizer
NLP:	Natural Language Processing
POS:	Part of Speech
QA:	Question Answering
QAS:	Question Answering System
RB:	Rule Based
TREC:	Text REtrieval Conference
URL:	Uniform Resource Locator

Chapter One: Introduction

1.1 Background

Information is very important in our day to day activity. Technology plays an important role in order to satisfy human beings information need through the use of Internet where people ask questions and a system provides an answer for their query. For instance, by using search engines a user submits a query and a web search engine displays a link to relevant web pages for each issued user's query, where the user is responsible for finding the correct answer from the listed links used for retrieving the documents. The documents with the same keyword with the query keyword are retrieved.

Natural language processing (NLP) is an automatic (or semi-automatic) processing of human language. It is a large and multidisciplinary field and closely related to linguistics. It also has links to research in cognitive science, psychology, philosophy, and maths (specially logic). Within Computer Science (CS), it relates to formal language theory, compiler techniques, theorem proving, machine learning, human-computer interaction and also related to Artificial Intelligence (AI) [1]. It is a collection of techniques used to extract grammatical structure and meaning from input in order to perform a useful task. As a result, natural language generation builds output based on the rules of the target language and the task at hand [2].

NLP has many applications such as: Information Retrieval, Information Extraction, Machine Translation, Text Summarization, and Question Answering. Question Answering System provides an answer for natural language questions rather than a linked list of documents.

The history of question answering system started in 1961 BASEBALL and in 1973 LUNAR QAS [3]. BASEBALL was a program for answering questions about baseball games played in the American league over only one season. Given a question such as "Who did the Red Sox lose to on July 5?" or "How many games did the Yankees play in July?" BASEBALL analysed the question, using linguistic knowledge, into a canonical form which was then used to generate a query against the structured database containing the baseball data. The second QAS was LUNAR designed "to enable a lunar geologist to conveniently access, compare and evaluate the chemical analysis data on lunar rock and soil composition that was accumulating as a result of the Apollo moon mission".

LUNAR could answer questions such as "What is the average concentration of aluminum in high alkali rocks?" or "How many Brescias contain Olivine?". The system was able to answer 90% of the in-domain questions posed by working geologists, without prior instructions as to phrasing.

Question Answering (QA) is a task in NLP that will automatically provide answers to questions posed in natural language [4]. It is an NLP based application which provides an exact answer for human beings question instead of a linked document, the sources of the answers can be a database or document collection (local or web). As a result, successful implementation of the QAS passes through document pre-processing, question analysis, document retrieval and answer extraction steps.

QA system can also be defined as a man machine communication device [5]. There are different question types such as acronym, counterpart, definition, biography, description, famous, stand for, synonym, why, name-a, name-of, where, when, who, what/which, how, yes/no and true/false. Where, when, which, yes/no, true/false, and name of are kinds of factoid questions where as definition, description, list, biography are non-factoid questions. Some questions are closed-domain (where the questions raised are in a specific domain such as in medicine) and open-domain which are questions almost about everything.

1.2 Motivation

TREC (Text Retrieval Conference) was started in 1992 as part of the TIPSTER (a program of research and development in the areas of information retrieval, extraction and summarization) text program. Its purpose was to support research within the information retrieval community by providing the infrastructure necessary for large scale evaluation of text retrieval methodologies, which have QA track amongst the tracks [36]. There are question answering systems developed for Amharic and other languages (non-factoid) and Afaan oromo factoid questions. The Amharic and other languages non-factoid question answering system can't help in answering the Afaan Oromo non-factoid question, as it needs different language dependent processing. The Afaan Oromo factoid question answering system can't answer the non-factoid question because, answer for factoid question are entity name and the non- factoid question require an explanation than named entities. This specific problem motivated us to study and investigate the possibilities of list, definition and description QA system for Afaan Oromo language.

1.3 Statement of the Problem

Afaan Oromo uses Latin based script called "Qubee" which has 37 basic characters. It is the official language of Oromia regional state of Ethiopia. In addition to this, the language is a medium of instruction in the schools of the region and studied as a subject in the Universities [12]. As a result, availability of Afaan Oromo textual information is highly increasing from time to time. So, literatures, a number of newspapers, magazines, educational resources, official credentials, and religious documents have been written in the language. This information can be found electronically on different online and offline sources of information in the language. The question and answering techniques differ from language to language due to word formation, grammatical arrangement and type of interrogative terms. Therefore, each question needs special consideration to return the correct answer according to their languages question answering techniques.

Since the emergence of the idea of question answering system, a lot of researches have been done worldwide on QA in English and other languages. In the case of Ethiopia, the field needs to be exploited more as Ethiopia is the home for speakers of more than 80 languages [12]. Some of the efforts that were made on the development of question answering for local language include Seid Muhie [6] (factoid question), Desalegn Abebaw [7] (factoid question), Tilahun Abdissa [18] (non-factoid questions) on Amharic and Aberash Tesfaye [22] on Afaan Oromoo factoid questions. The problem that this research work tries to address is how to develop Afaan Oromo non-factoid question of type list, definition and description.

1.4 Objectives

1.4.1 General Objective

The general objective of the study is to develop Afaan Oromo Question Answering System for definition, list and description question types.

1.4.2 Specific Objectives

The specific objectives of this research work are:

- To review related works in Afaan Oromo and other languages.
- To extract specific features of Afaan Oromo definitional, list and description questions and answers.
- To prepare Afaan Oromo corpora and testing questions

- To design the architecture of AOLDDQAS.
- To analyze question and answer patterns on list, definition and description questions
- To develop an algorithms for AOLDDQAS.
- To develop a prototype of the system.
- To evaluate the developed prototype using evaluation metrics.

1.5 Scope and Limitations

Naturally, question answering is a very complex task which needs understanding of natural language techniques. A full-fledged QA system will require a number of natural language processing tools such as sentence parser, chunker, part of speech (POS) tagger, stemmer, named entity recognizer (NER), word net and so on. Even though some of the NLP tools have been developed by some researchers, they are not publicly available for integrating with a system. This research focuses on: Afaan Oromo non-factoid question of type list, definition and description questions. Non-factoid questions other than list, definition and description questions are out of our scope.

1.6 Methodology

Literature Review

In order to understand the Afaan Oromo language structure and QA system, literatures will be reviewed. Furthermore, discussions will be held with linguistics. The data set will be prepared by collecting Afaan Oromo list, definition and description questions and answers.

Data Sources

Data will be collected from different websites, journals, educational books and so on for understanding the characteristics of question types and their respective answers.

Development Tools

In order to develop the Afaan Oromo List, Definition, Description Question Answering (AOLDDQA) system, as a developmental tool, Java programming language will be employed as a major developmental tool for the prototype, Apache Lucene [32] will be used for indexing and relevant document retrieval task, a rule based method will be

implemented for the question classification task and stemmer for stemming [15] and morphological analyzer [25] for lemmatization will be used.

Evaluation

The performance of AOLDDQA system will be done by collecting list, definitional and descriptional questions and evaluate the system‟s performance against manual answers. We will use percentage, precision, recall and F-Score as an evaluation method.

1.7 Application of Results

As QA provides precise answers to a given natural language question, the AOLDDQA system can be applicable in finding answers for Afaan Oromo list, definition and description type of questions from a collection of documents and also can be used for teaching, learning, and research.

1.8 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 presents literature review in which different concepts related to the thesis are presented. Chapter 3 is about works somehow related to our work which are done by other researchers in Amharic, Afaan Oromo and other languages. Chapter 4 deals with the detailed design and implementation of the system. Chapter 5 deals with the experiments done in every component and the results achieved together with explanations of how such results happen. Chapter 6 winds up our work by presenting a conclusion and future works recommendation for the improvement of the system.

Chapter Two: Literature Review

In this chapter we will concentrate on addressing Question Answering (QA) system development strategies. The first section presents Information Retrieval (IR). The next section will cover general architectures and discusses particularly on techniques and approaches in question analysis, document retrieval, document analysis and answer extraction components of a QA system. The third section discusses about Morphological Analysis. The fourth section discusses the Lucene API. Finally, we will discuss about Afaan Oromo Language.

2.1 Information Retrieval

As stated by Manning [31], Information Retrieval (IR) is finding material of an unstructured nature that satisfies an information need from within large collections. It is concerned with searching of documents for information from document corpus and the World Wide Web. IR searches both structured and unstructured information and it includes various process and techniques. The whole IR system includes three main subsystems:

- Indexing: is an offline process of extracting index terms from document collection and organize them using indexing structure to speed up searching [12]. The most common indexing structure for text retrieval is the inverted file. This structure is composed of two elements: the vocabulary and the term occurrences. The vocabulary is the set of all words in the text. For each word in the vocabulary, a list of all the text positions where the word appears is stored. The set of all those lists is called occurrences. It is language dependent process which varies from language to language.
- Processing: case normalization, stop word removal, stemming, lemmatization are applied on users query. In the case of textual retrieval, query terms are generally pre-processed by the same algorithms used to select the index objects. Additional query processing (e.g., query expansion) requires the use of external resources such as thesauri or taxonomies.
- Searching and Ranking: user queries are matched against information items. As a result of this operation, a set of potential information items is returned in response to user needs. The ranking step aims to predict how relevant the items

are comparatively to each other, thus returning them by decreasing the order of estimated relevance.

One issue regarding information retrieval systems is the issue of predicting which documents are relevant to the user queries. Such decision is dependent on the ranking algorithm, which orders the retrieved documents in some order, the system uses. This section will cover three of the most important IR models, namely [9]:

- Boolean: documents and queries are represented as a set of index terms.
- Vector space: documents and queries are represented as vectors in a t-dimensional space.
- Probabilistic: documents and queries representations are based on probability theory.

A. Boolean Model

The Boolean Model is a simple retrieval model based on set theory and Boolean algebra [33]. The model represents documents by a set of index terms, each of which is viewed as a Boolean variable and valued as true, if it is presented in a document. Queries are represented as a Boolean expression composed by index terms and logic operators AND (product), OR (sum), and NOT (difference). The logical operator AND is used to group set of terms in to single query/statement. For example „Information AND Technology" is two term query combined by „AND". In such case only document indexed with both terms will be retrieved. If terms in the user query are linked by operator OR, documents with either of terms or all terms will be retrieved. For example, if query is Information OR Technology, document containing Information, or Technology, or Information Technology will be retrieved.

What makes Boolean model good model is that it creates a sense of control to expert/user over the system. It is the user who is in charge for deciding what should or shouldn„t be retrieved. Query reformulation is also simple because user is in charge of deciding what should be retrieved and should not. In contrast, Boolean model may not retrieve anything if there is no matching document or, retrieves all documents if terms in query are matching with it. It is simple but not efficient [24].

B. **Vector Space Mode**

The vector space model is based on algebraic concepts, represents the documents and queries as vectors in a multidimensional space, whose dimensions are the terms used to build an index to represent the documents [24]. Query and document similarities can be compared by calculating their vectors using the cosine similarity measure. The cosine similarity between the document and the query is calculated by the following formula:

$$\text{cosine similarity}(d, q) = \frac{V_d . V_q}{|V_d||V_q|}$$

Where, d represents term of the document and q represents terms of the query, v_d is vector in the document direction and v_q is the vector in the query direction.

If terms of the query occur in the document, the value of the cosine similarity is non-zero. A cosine value of zero means the query term does not exist in the document being considered.

C. **Probabilistic Model**

The probabilistic retrieval model is based on the Probability Ranking Principle, which states that an information retrieval system is supposed to rank the documents based on their probability of relevance to the query, given all the evidences available. The principle takes into account that there is uncertainty in the representation of the information need and the documents. The rank of the documents is given by the following ratio:

$$similarity(d, q) = \frac{P(R/d)}{P(\bar{R}/d)}$$

Where, P(R/d) is the probability that a document d is relevant to a query q and P($\bar{R}$/d) is the probability that a document d is non-relevant to a query q.

Information extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. Where, the focus is on the recognition, tagging and extraction of certain key elements of information (e.g persons, companies, locations, organizations, etc.) from large collections of text into a structured representation.

2.2 Question Answering

A question answering (QA) is a task that aims to automatically give answers to questions described in natural language [9]. It allows users to have exact answer rather than having list of potentially relevant documents. The traditional search engine focuses on retrieving

related documents and returns list of related documents for the users and users must scan to get the necessary information. Whereas, QA system answers the question in the form of exact answer which is extracted from source documents. QA system needs more complex natural language processing (NLP) tools for precisely understanding the user‟s intention as well as to extract correct answers. But, in the case of IR, a simple technique is sufficient to return content-rich documents. In recent time, the automatic question answering system has become an interesting research field and resulted in a significant improvement in its performance which has been largely driven by the TREC (Text Retrieval Conference) QA Track [26].

A question answering system is either factoid or non-factoid questions. Factoid questions return answers in the form of a name of a person, name of a country, name of an organization, quantity of something and date or time on which something happens. On the other hand, non-factoid questions are questions that ask for definitions, reasons, biography, methods, and procedures. The answers for non-factoid type of questions are more complex than factoid questions.

A question answering has many applications. We can subdivide these applications based on the sources of the answers. The sources can be structured data (databases), semi-structured data (for example, comment fields in databases) and unstructured (free text) corpora. We can also distinguish between domain independent question answering systems (systems designed to answer general questions in all domains) and domain specific systems (systems designed to answer questions generated only within a certain domain like medicine, chemistry, and so on).

2.3 General Architecture of QAS

Currently, there are dozens of textual question answering systems described in the literature. In 2002, 34 research groups participated in the question answering track of the annual Text REtrieval Conference (TREC), each group having implemented their own system [8]. These systems cover a wide spectrum of different techniques and architectures which are impossible to capture all variations within a single architecture. Most of the time, question answering system has four basic components such as question analysis, document retrieval, passage retrieval and answer extraction.

The question analysis component is responsible to analyze questions that determine the expected answer type. In addition, it is responsible to formulate proper queries for document retrieval. The document retrieval module will use the queries formulated by the question analysis to retrieve the top N relevant documents. The answer extraction component takes as input documents that are likely to contain an answer to the original question from document retrieval component, which are sent to the answer selection component. Finally, the answer selection component selects the phrase that is most likely to be a correct answer from a number of phrases of the appropriate type, as specified by the question analysis component [21]. Figure 2.1 shows a pipelined architecture of a question answering system taken from [21]. The figure outlines the major components in a question answering system.

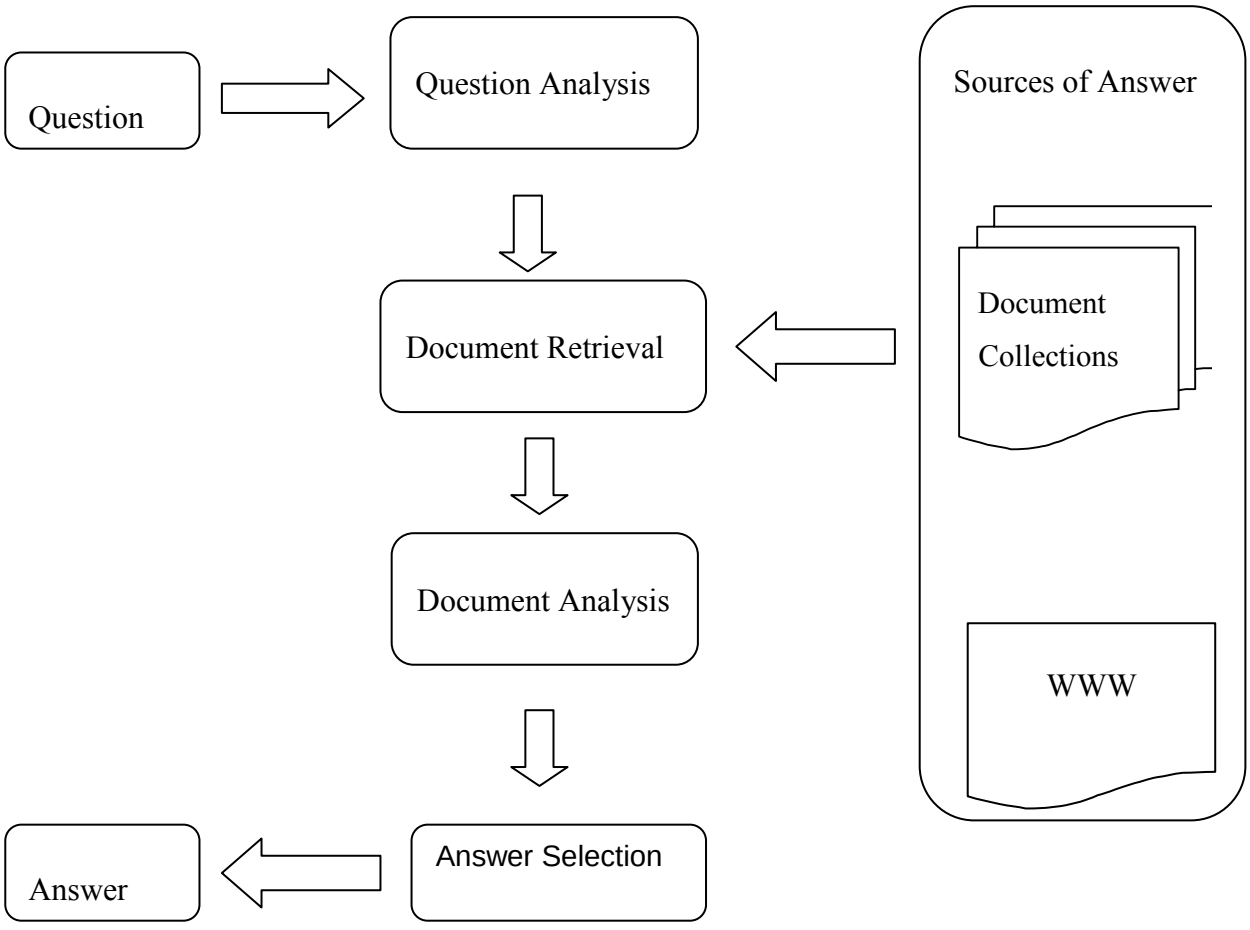

Figure 2.1, *General Architecture of QAS*

2.3.1 Question Analysis

Question analysis is the process of constructing representation of questions, deriving of expected answer types, and extracting of keywords [28]. In the question analysis stage,

the type of question will be analyzed. The question type further illustrates what will be the expected answer type. It is the question analysis stage that is also responsible for constructing proper query for the IR component of the QA system. Correctly identifying the expected answer type will help the later stage of answer extraction to correctly identify answers. Therefore, wrong question analysis means that the document retrieval component will retrieve wrong documents as well as the answer extraction component will extract wrong answer or no answer. The question analysis component has two sub-components: question classification and query generation.

A) Question Classification

A study of question classification plays an important role for the question answering system. A question classification technique helps in order to answer a user question correctly by identifying user‟s need, which simplifies the searching of an answer by giving clues. The question classification task predicts the type of the answer based on the provided user‟s query.

To identify the type of a question, the question is classified in a number of ways. Currently, researchers use different approaches such as Machine Learning, Ruled Based and Language Model.

B) Query Generation

The other task of the question analysis component in question answering system is formulating appropriate query from the user‟s natural language questions so that it would be sent to the document retrieval component of the system. First, the characters of the question are normalized, short words are expanded, and stop words are removed, then the query generator removes the interrogative terms that are found within it in order to increase matching of the query to the relevant documents which contain the actual answer to the user‟s question. The question particles help in the construction of question sentences. These words are not important for the retrieval of relevant documents. Thus, question particles are needed to be removed in the query generation process. The goal of query generation is to improve the overall quality of the ranking of the documents returned in response to user‟s query [20].

2.3.2 Document Retrieval

The function of the document retrieval component is not to find actual answers to the question, but to identify documents that are likely to contain an answer [21]. Document

retrieval aim to return relevant documents to a user‟s query, where the query is a set of keywords. A document is considered relevant if its content is related to the query [19]. The main purpose of the document retrieval component is to select an initial set of candidate answer-bearing documents from a large text collection prior to sending them to a downstream answer extraction module. In an effort to pinpoint relevant information more accurately, the documents are split into several passages, and the passages are treated as documents. Thus, many QA systems also have a passage retrieval stage, interposed between the document retrieval and answer extraction components, which can be thought of as a second, smaller scale IR module. Using a passage-based retrieval approach instead of a full-document retrieval approach has the additional advantage that it returns short text excerpts instead of full document which are easier to process by later components of the question answering system. Document retrieval has a long tradition and many frameworks have been developed over the years, resulting in sophisticated ways to compute the similarity between a document and a query. Depending on the retrieval engine that is actually used, the retrieval component returns either an unordered set of documents that are likely to contain an answer, or a ranked list of documents, where the documents are ranked with respect to their likelihood of containing an answer. Document retrieval effectiveness is critical to the overall performance of a question answering system. If the document retrieval component fails to return any document that contains an answer, even optimally functioning answer extraction and answer selection components will inevitably fail to return a correct answer to the user [21].

2.3.3 Document Analysis

Candidate answers are located in small part of a document [29]. Sentence/Passage extraction can be performed by segmenting each document into small sentence/passage and selects suitable sentence/passage related to keywords. In segmenting the set of relevant documents, in order to detect sentence in a document, punctuation marks can be used as separators. In detecting paragraphs of a document, empty lines can be used as separators. So, in this way from the set of candidate documents the set of candidate sentences/passages which are supposed to contain the candidate answers are retrieved.

Once candidate answer-bearing documents or document passages/segments have been selected, these text segments may then be further analyzed. The document analysis component searches through the documents returned by the retrieval component to identify phrases that are of the appropriate type, as specified by the question analysis

component. Typically, however, systems now analyses the selected documents or document portions using at the very least a named entity identifier, which recognizes and classifies multiword strings as names of includes person, organization, dates, locations, temporal and spatial distances, etc.

At this stage, there are a number of ways to further analyze documents. Such as sentence splitting, part-of-speech tagging, and chunk parsing. In order to establish an explicit link between a phrase of the appropriate type and the question, the syntactic structure, pattern matching, or lexical chaining, then linear proximity is often used.

2.3.4 Answer Selection

At this stage, question answering systems are faced with a set of text fragments which are possible correct answers. These candidates usually originate from different passages and are often extracted using different strategies. Moreover, these textual fragments may not always constitute full answers. The set of answer candidates obtained through answer extraction could include [35]:

Incorrect: the answer string does not contain a correct answer.

Not Supported: the answer string contains a correct answer but the document returned does not actually answer the question.

Not Exact: the answer string contains a correct answer and the document supports that answer, but the string contains more than just the answer or is missing bits of the answer.

Locally Correct: the answer string consists of exactly a correct answer that is supported by the document returned, but the document collection contains a contradictory answer that the assessor believes is better.

Globally Correct: the answer string consists of exactly the correct answer, that answer is supported by the document returned, and the document collection does not contain a contradictory answer that the assessor believes is better.

Answering definitional, biographical, how, why and other complex questions require to put together partial answers from different documents, in which this task is handled by the answer generator. So, answer generation is about taking a candidate answer and produce correct and complete answers with corresponding confidence scores. This task

involves combining evidence from several answer candidate components in order to generate meaningful correct answers with high confidence scores [31].

2.4 Morphological Analysis

Morphology is the study of word formation – how words are built up from smaller pieces [17]. Morphological analysis is the segmentation of words into their component morphemes and the assignment of grammatical morphemes to grammatical categories and lexical morphemes to lexemes [25]. When we do morphological analysis, then, we"re asking questions like, what pieces does this word have? What does each of them mean? How are they combined? It is the study of how words are composed of morphemes (the smallest meaning-bearing units of a language).

There are two types of morphology: inflectional and derivational. Inflectional morphology is concerned with the inflectional changes in words where word stems are combined with grammatical markers for things like person, gender, number, tense, case and mode. Inflectional changes do not result in changes of parts of speech. On the other hand, derivational morphology deals with those changes that result in changing classes of words (changes in the part of speech). For instance, a noun or an adjective may be derived from a verb.

A morpheme is the smallest semantically meaningful unit in a language. It is not identical to a word, and the principal difference between the two is that a morpheme may or may not stand alone, whereas a word, by definition, is a freestanding unit of meaning. Every word comprises one or more morphemes. We define different kinds of morphemes based on various properties like where they show up in words. Morphemes can be classified in two ways: **free versus bound morphemes and roots, affixes versus combining forms**

Free morphemes are morphemes that can stand alone as words.

Examples: muca (toddler), hanga (amount), fula (face)

Bound morphemes are morphemes that cannot stand on their own as a word, but rather must be attached to a free morpheme whenever you say it.

Examples: hin-*, -f, -ef,*

Root morphemes are the primary piece of meaning in a word, to which affixes can be added. Examples *kuf- in kufte, kufe,kufu* where, kuf- is the root word for kufe, kufte, kufu.

Affix morphemes are morpheme which attaches to roots (or stems), changing their meaning in regular ways. The affixes can be prefix, suffix and infix. The first and the second types of affixes occur at the beginning and at the end of a root respectively in creating a word whereas; infix is a morpheme inserted within morpheme. Moreover, the work of [15] shows that Afan Oromo does not have infixes like English.

Example: In "dhugaatii", -aatii is a suffix and dhug- is a stem

In "hindeemu", hin-is a prefix,-deem- is a root and -u is a suffix.

2.5 Lucene

Lucene is an open source information retrieval software library written in Java, used to build and search indexes [23]. Lucene can index any text-based information we like and then find it later based on various search criteria. Although Lucene only works with text, there are other add-ons to Lucene that allow us to index Word documents, PDF files, XML, or HTML pages. It provides a basic framework that we can use to build full-featured search into our application. The following explanation is adopted from [23].

2.5.1 Indexing

Indexing is a way of creating cross-reference lookup (index) in order to facilitate searching. Since Lucene's index lists the documents that contain a term, it falls into the family of indexes known as an inverted index. Inverse document frequency reflects how frequent a term is in the whole collection. The underlying principle is that a term that appears in a few documents gives more information than a term that appears in many documents. This means a term that appears in many documents has a weak discriminating power to select relevant documents over a document collection [28].

Before indexing documents in Lucence, index pre-processing operations (character normalization, stop-word removal, short word expansion, and stemming or morphological analysis) are applied. IndexWriter can‟t index text unless it‟s first been broken into separate words, using an Analyzer.

As stated in [23], Lucene indexer has IndexWriter, Directory, Analyzer, Document, and Field classes for performing indexing procedure.

- Index writer: creates a new index or opens an existing one, and then adds, removes or updates documents in the index.
- Directory: represents the location of a Lucene index. It‟s an abstract class that allows its subclasses to store the index. IndexWriter uses FSDirectory or RAMDirectory, and creates the index in a directory in the file system.
- Analyzer: Before text is indexed, it‟s passed through an analyzer, performs pre-processing of the data. The analyzer class is language dependent as the pre-processing operations are language specific.
- Document: is simply a container for multiple fields, which is the class that actually holds the textual content to be indexed. It can be considered as a virtual document a chunk of data, such as a web page, an email message, or a text file that we want to make retrievable at a later time.
- Field: represents the document or metadata associated with that document. The metadata such as author, title, subject, date modified, and so on, are indexed and stored separately as fields of a document.

2.5.2 Searching

Searching is the process of looking for words in the index and finding the documents that contain those words. As stated in [23], Lucene searcher has: Index Searcher, Term, Query, Term Query and Top Docs classes.

- Index Searcher: is used to search from the index
- Term: is the basic unit for searching. Similar to the field object, it consists of a pair of string elements: the name of the field and the word (text value) of that field.
- Query: is the common, abstract parent class.
- Term Query: is the most basic type of query supported by Lucene, and it‟s one of the primitive query types. It‟s used for matching documents that contain fields with specific values.
- Hits - Hits class contains the Document objects that are returned by running the Query object against the index.

2.6 Afaan Oromo Language

Afaan Oromo is one of the major African languages that is widely spoken and used in most parts of Ethiopia and some parts of neighboring countries like Kenya and Somalia. Currently, it is an official language of Oromia regional state. It is used by Oromo people, who are the largest ethnic group in Ethiopia, which amounts to 40% of the total population [15]. With regard to the writing system, Qubee (a Latin-based alphabet) has been adopted and become the official script of Afaan Oromo since 1991. Now, it is language of public media, education, social issues, religion, political affairs, and technology.

2.6.1 Afaan Oromo Writing System

Afaan Oromo is a phonetic language, which means that it is spoken in the way it is written. The writing system of the language is straightforward which is designed based on the Latin script. Unlike English or other Latin based languages, there are no skipped or unpronounced sounds/alphabets in the language. Every alphabet is to be pronounced in a clear short/quick or long /stretched sounds. In a word where consonant is doubled the sounds are more emphasized. Besides, in a word where the vowels are doubled the sounds are stretched or elongated [30].

Like in English, Afaan Oromo has vowels and consonants. Afaan Oromo vowels are represented by the five basic letters such as a, e, i, o, u. Besides, it has the typical Eastern Cushitic set of five short and five long vowels by doubling the five vowel letters: „aa‟, „ee‟, „ii‟, „oo‟, „uu‟[30].

Consonants, on the other hand, do not differ greatly from English, but there are few special combinations such as “**ch**” and “**sh**” (same sound as English),“**dh**” in Afaan Oromo is like an English "*d*" produced with the tongue curled back slightly and with the air drawn in so that a glottal stop is heard before the following vowel begins. Another Afaan Oromo consonant is “**ph**” made when with a smack of the lips toward the outside, “**ny**” closely resembles the English sound of “gn”. We commonly use these few special combination letters to form words. For instance, **ch** used in **barbaachisaa** *‚important’*, **sh** used in **shamarree ‘***girl’*, **dh** use in **dhadhaa** *‚butter’* , **ph** used in **buuphaa ‘***egg’,* and **ny** used in **nyaata** *‚food’*[12] .

In general, Afan Oromo has 37 letters (32 consonants and 5 vowels) called "**Qubee".** In general, all letters in English language are also in Afan Oromo except the way it is written.

2.6.2 Punctuation Marks in Afaan Oromo

Punctuation is placed in text to make meaning clear and reading easier. Analysis of Afaan Oromo texts reveals that different punctuation marks follow the same punctuation pattern used in English and other languages that follow Latin writing system. Similar to English, the following are some of the most commonly used punctuation marks in Afaan Oromo.

i. Tuqaa *Full stop* (.): is used at the end of a sentence and in abbreviations.
ii. Mallattoo Gaafii *Question mark* (?): is used in interrogative or at the end of a direct question.
iii. Rajeffannoo *Exclamation mark* (!): is used at the end of command and exclamatory sentences.
iv. Qooduu *Comma* (,): it is used to separate listing in a sentence or to separate the elements in a series.
v. Tuqlamee *colon* (:): is used to separate and introduce lists, clauses, and quotations, along with several conventional uses, and etc.

2.6.3 Afaan Oromo Language Part of Speeches

The Afaan Oromo language words can be categorized into nouns, verb, adverb, adjective, pronoun and prepositions.

I. Nouns

A noun is a word that helps to identify the categories of things, people, places and ideas. Nouns in Afaan Oromo are inflected for gender, definiteness and number.

i. Gender

Afaan Oromo has a two gender system (feminine and masculine). The language uses **-ssa** for masculine and **-ttii** for feminine.

Obboleessa	*brother*	- obboleettii	*sister*
Ogeessa	*expert* (m.)	- ogeettii	*expert* (f.)

Names of astronomic bodies and geographical terms such as sun 'aduu', star 'urjii', town 'magaala', river 'laga' are feminine. It is due to this fact that there are different subject forms for the noun **biyya** ‚*country'*. Example: **biyy**i(*m.)* or **biitti** (*f.*).

ii. Number

Afaan Oromo has different suffixes to form the plural of a noun. The use of different suffixes differs from dialects to dialects. According to[30], the suffixes that can be added to form plural nouns are: -oota, -een, -lee, -an, -wwan, -olii .

For example: hiriyaa – hiriyoota (friend- friends)

gaangee - gaangolii (mule- mules)

iii. Definiteness

Afaan Oromo language does not possess a special word class of articles. Instead, demonstrative pronouns are used to express definiteness.

kitaabni kun *this/ the book (Subject)* kitaabni sun *that/ the book (Subject)*

kitaaba kana *this/ the book (Object)* kitaaba sana *that / the book (Object)*

iv. Derived noun forms

Afaan Oromo is very productive in word formation by different means. The most common word formation methods are derivational and compounding [31].

a. Derivation

Derivational suffixes are added to the root or stem of the word. The following suffixes play an important role in Afaan Oromo word derivation. They are **-eenya, -ina, -ummaa, -annoo, -ii, -ee, -a, -iinsa, -aa, -i(tii), -umsa, -oota, -aata,** and **–ooma.**

Examples: jabaa *strong* jabeenya *strength*

jabina *strength, hardness* jabee *intensive*

b. Compound words

On the other hand, it seems that the use of genitive constructions is a very old method of forming compound nouns, as traditional titles show.

abbaa gadaa - *traditional Oromo president*

abbaa caffee - *chairman of the legislative assembly*

II. Verbs

Verbs are content words that denote an action, occurrence, or state of existence. Afaan Oromo has base (stem) verbs and four derived verbs from the stem. Moreover, verbs in Afaan Oromo are inflected for gender, person, number and tenses. There are four derived stems, the formation of which is still productive, Autobenefactive, Passive, Causative and Intensive.

a. Autobenefactive

The Afan Oromo autobenefactive (or "middle" or "reflexive-middle") is formed by adding **-(a)adh**, **-(a)ach** or -(a)at or sometimes **-edh**, **-ech** *or* –**et** to the verb root. This stem has the function to express an action done for the benefit of the agent himself.

Example:

bitachuu **-** *to buy for oneself* the root verb in this case is **bit-**

b. Passive

The Oromo passive corresponds closely to the English passive in function. It is formed by adding **-am** to the verb root. The resulting stem is conjugated regularly.

Example: beek- *know* beekam- *be known*

c. Causative

The Afaan Oromo causative of a verb corresponds to English expressions such as 'cause ', 'make ', 'let '.With intransitive verbs, it has a transitiving function. It is formed by adding **-s, -sis, or -siis** to the verb root.

Example: deemuu - to go deemsisuu **-** to cause to go

d. Intensive

It is formed by duplication of the initial consonant and the following vowel, geminating the consonant.

Example:

Waamuu **-** *to call, invite* wawwaamuu **-** *to call intensively*

III. Adjectives

An adjective is a word which describes or modifies a noun or pronoun. A modifier is a word that limits, changes, or alters the meaning of another word. Unlike English, adjectives are usually placed after the noun in Afaan Oromo. For instance, in (Tolaan farda adii bite) "*Tola bought white horse"* the adjective **adii** comes after the noun **farda**. Moreover, in Afaan Oromo sometimes it is difficult to differentiate adjective from noun.

Example: dhugaa - *truth, reality, true, right*

dhugaa keeti - *your truth/ you are right* (truth served as noun)

i. Gender

In Afaan Oromo, adjectives are inflected for gender. We can divide adjectives into four groups with respect to gender marking. These are:

a. In the first group, the masculine form terminates in –**aa**, and the feminine form in –**oo**.

Example: guddaa (*m*.) nama guddaa - *a big man*

guddoo(*f*.) nama guddoo - *a big woman*

b. In the second group, the masculine form terminates in –**aa**, the feminine form in – **tuu** (with different assimilations).

Example: dheeraa(*m.)* nama dheeraa - *a tall man*

dheertuu**(***f.)* intal dheertuu - *a tall girl*

c. Adjectives that terminate in –**eessa** or **–(a)acha** for masculine and have a feminine form in –**eettii** or –**aattii**.

Example: dureessa (*m.)* nama dureessa - *a rich man*

dureettii **(***f.)* nitii dureettii - *a rich woman*

d. Adjectives whose masculine form terminates in a long vowel other than **–aa** as in short vowel **–a** (but not of the suffix –**eessa**/-**aacha**) are not differentiated with respect to their gender.

Example: colee(*m*.) farda collee - *an active horse*

colee(*f.)* gaangee collee - *an active mule*

ii. Number

There are four groups of adjectives with respect to number. These are:

a. Most of the adjectives form the plural by reduplication of the first syllable masculine and feminine adjectives differ in plural as they do in singular:

Example:	Singular	Plural
	guddaa**(***m***.)**	gugguddaa(*m.)*
	guddoo**(***f***.)**	gugguddoo(*f.)*

b. There is a further plural form which is gender neutral for adjectives of this group beside a special masculine and feminine plural. This plural form terminates in **-oo**, and is sometimes used with reduplication and sometimes without. Table 2.1 shows examples of plural adjectives formed by reduplication which are gender neutral.

Table 2.1: *Examples of gender neutral adjectives*

Singular		Plural		Plural
M	F	M	F	Neutral
Dheeraa Jabaa	Dheertuu Jabduu	Dhedheeraa Jajjabaa	Dhedheertuu Jajjabduu	Dhedheertuu Jajjaboo

c. Adjectives which may function as nouns as well form the plural only by using noun plural suffixes. Table 2.2 shows examples of plural adjectives formed using noun plural suffixes

Table 2.2: *Examples of plural adjectives*

Singular		Plural	
M	F	M	F
Dureessa	Dureettii	Dureeyyii/dureessota	Dureettiwwan

d. Adjectives of the fourth group form the plural without marking the gender, very often by reduplication of the first syllable. Sometimes adjectives of this group form the plural by using a noun plural suffix. Table 2.3 shows examples of plural adjectives formed by reduplication of the first syllable or using noun plural suffixes.

Table 2.3: *Examples of plural adjectives formed plural suffixes*

Singular	Plural	English
Adii	a`adii/adaadii	White
Colee	Colleewwan	Active

iii. Definiteness

The demonstrative pronouns that express definiteness in Afaan Oromo follow the adjective if the noun is qualified by an adjective and a demonstrative pronoun as well.

Example: Namicha dheeraa sana argitee? *Did you see that tall man?*

The suffix **–icha** that sometimes has a definite function normally is suffixed to nouns, but it can be suffixed to adjectives or numerals, too,

Example: lagni guddicha - *the big river* namicha tokkicha - *a single man*

iv. Compound adjectives

In the new terminology of Afaan Oromo compound, adjectives play a growing role.

Example: afrogaawaa - *afur + rogaawaa* sibilala - *sibila + ala*

IV. Adverbs

Adverbs have the function to express different adverbial relations such as relations of time, place, and manner or measure.

Some examples of adverbs of time:

amma - *now,* booda - *later*

Some examples of adverbs of place:

achi(tti) - *there ,* ala - *outside*

Some examples of adverbs of manner:

saffisaan - *quickly,* sirritti - c*orrectly*

Some examples of adverbs of measure:

baay"ee , danuu - *much , many , very,* duwwaa - *only, empty*

V. Pre- Post and Para-Positions

i. Postpositions

Postpositions can be grouped into suffixed and independent words.

a. Suffixed postpositions (**-tti** *in, at, to,* **-rra/irra** *on,* **-rraa/irraa** *out of, from*)

Example: Adaamaatti yoom deebina? - *When shall we go back to Adama*?

Gammachuun sireerra ciise. - *Gemechu lay down on bed.*

b. Post position as independent words

(**ala** *outside,* **wajjiin** *with , together with,* **bira** *beside,* **teellaa** *behind*)

Example:

Namoota nu bira jiraniis hin jeeqnu. - *We don't hurt people who are with us.*

ii. Prepositions

(akka - *like,according to,* gara - *to, in the direction of,*

hanga/hamma - *until, up to*)

Example: Namni akka harkaan waa hojjechuuf fayyadamu argi maalitti fayyadamaa? *As people use hands to work something what does the elephant use?*

iii. Para-positions

(Gara… tti *to,* Gara… tiin - *from the direction of*)

Example: Lukkichi rifatee jeedaloo dheesuuf gara manaatti gale. **(***The cock was scared and went home to take refuge from the fo)x.*

VI. Conjunctions

Conjunctions are unchanging words which coordinate sentences or single parts of a sentence. The main task of conjunctions is to be a syntactical formative element that establishes grammatical and logical relation between the coordinated constituents. According to [15], the main functions of conjunctions are indentified 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

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)*

iv. Conjunctions consisting of several segments

Conjunctions consisting of several segments are copulative or disjunctive conjunctions which –as they stand separately from each other –are to emphasize the segments of a

parallel construction. These are stable, stereotyped constructions the first segment of which has to be followed by a certain second segment:

Example: **–s… -s -** *as well as*

Jechoota hudhaa wajjiiniis, hudhaa malees karaa lamaan barreeffaman *(Words with glottal stop as well as without glottal stop are written in two ways).*

2.6.4 Afaan Oromo Abbreviations

Abbreviations are mostly formed by taking initial letters of multiword sequences to make up a new word. Sometimes, they can be formed from initial and non-initial letters.

In Afaan Oromo, abbreviations are used to represent dates A.L.I (Akka Lakkoofsa Itiyoophiyaa) to mean in Ethiopian calendar, A.L.A (Akka Lakkoofsa Awurooppaa) to mean in Gregorian calendar, months and dates by short words. 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).

2.6.5 Questions in Afaan Oromo

In forming question statements, different 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.

In the English language, interrogative articles such as ***who*, *what, where, when, why, how*** are used to construct a questions. In the same way, Afaan Oromo interrogative particles help to construct a question sentence. Interrogative particles are also known as interrogative pronouns. 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 the factoid and non-factoid questions.

Chapter Three: Related Work

This chapter presents a review of question answering system of the local and foreign languages. The first two sections present QA system on local languages the Amharic Non-factoid QA system and Afaan Oromo factoid QA. The other sections describe works on foreign languages. The next three sections discuss about English, Arabic and Japanese Non-factoid QA. The next section will cover the Singapore Definitional QA. The last two sections present the Portugese and English list QA system. Finally, the last section will give a summary of the reviewed related works.

3.1 Amharic Non-Factoid Question Answering

The work in [18] presents Amharic Question Answering System for non-factoid questions. The researchers developed Amharic question answering system for definitional, descriptional and biographical question types. This QAS consists of four major components, document pre-processing, question analysis, document analysis and answer extraction.

The first component in this paper, document pre-processing component, performs character normalization, short word expansion, stop word removal, stemming, and lemmatization. The question analysis component has subcomponents which perform question classification, query generation and query expansion tasks. In classifying the natural language question, the question classification sub component has implemented rule based and machine learning algorithms. In this research, the rule based classifier out performs the SVM. The document retrieval component retrieves and filters candidate documents which are relevant to the answer extraction component of the system. Finally, the answer extraction component selects the best answer by using the manually written rules for definition and description question classes and for the biography question types the summarizer is implemented in the extraction of the answers.

3.2 Afaan Oromo Question Answering

The work in [22] presents a factoid type QAS for Afaan Oromo language. The researchers developed a QAS for Afaan Oromo factoid questions (person, place, number and time). This QAS consists of four major components, document pre-processing, question analysis, document retrieval, and answer extraction.

In this paper, the first component is pre-processing where the documents and user queries are pre-processed using tokenization, short word expansion, and stop word removal, case

normalization, stemming and indexing. The paper describes, indexing process is excluded from query pre-processing. The question classification and query generator are subcomponents of the question analysis component used for classifying user queries and generating keywords used as an input in the document retrieval component. The researchers used two models for query classification, rule based and machine learning model known as Support Vector Machines (SVM) and the evaluation shows that the rule based question classifier model out performs the SVM based classifier model. The third component is document retrieval, the Lucene API open source search library for the retrieval of relevant documents were implemented. The answer extraction component used Named Entity Recognizer (NER) and pattern matching for extracting answers from the retrieved documents.

3.3 English Non-Factoid Question Answering

The work in [28] introduced a QA scheme that answers definitional questions of the form "what is X?" and "who is X?". The system first finds the target term (the concept for which information is being sought). A simple pattern-based parser was used to extract the target term using regular expressions. If the natural language question did not fit into any of the patterns, the parser heuristically extracts the last sequence of capitalized words in the question as the target. Then, nuggets relevant to the target term are extracted using database lookup, web dictionary lookup, and document lookup technique. Finally, answers from the different sources are merged to produce the final output. The target, the pattern type, nugget and source sentence are stored in a relational database. Then, the database lookup technique answers definitional questions simply by looking for relevant nuggets in the database using the target terms as query. The dictionary lookup defines questions using Merriam Webster online dictionary. Keywords from the target terms and the target itself are used as the query to Lucene IR engine. Then, filters and tokenizes the top one hundred documents into sentences and scored each based on their keyword overlap and inverse document frequency. The document lookup technique employs traditional document retrieval to extract relevant nuggets if no answers are found by the other two techniques. Finally, the answer merging component merges results from all the three sources. A simple heuristic was used to avoid redundancy, i.e., if two responses share more than sixty percent of their keywords, then one of them is randomly discarded.

The work in [16] presents a QA system known as QUANTICO is a cross-language open domain question answering system developed for both English and German factoid and definition question. It uses a common framework for both monolingual and cross-language scenarios, but with different configurations for each type of question (definition or factoid) and different workflow settings for each task. The researchers used five components, question analysis, translation services and alignment, passage retrieval, answer extraction and answer selection.

In question analysis, they used local lexico-syntactic criteria for determining the parts of information (*a-type*, *q-type, q-focus* and *q-scope)* tags. In translation service and alignment component, a-priori and posteriori method were used. Queries are translated online. For retrieving passages for factoid questions named entity and for definition questions some structural linguistic patterns are used with explanatory and descriptive purposes. Then, the answer extraction component extracts answer for factoid questions named entities and for definition question passages attained by matching them against a lexico-syntactic pattern. The answer selection component sorts out a list of top answers based on a distance metric defined over graph representations of the answer‟s context.

The work in [10] describes a QA system for English known as DefScriber that answers definitional questions of the form "What is X?" using goal driven and data driven methods. The main stages in DefScriber operation are input, document retrieval, predicate identification, data-driven analysis, and definition generation.

The document retrieval module uses a fixed set of patterns to identify the term to be defined in the question, and then generates a set of search queries. These queries are sent to a web search engine until the specified number of documents is retrieved. Once documents are retrieved, goal-driven analysis is performed to identify predicates. The system examines documents for the instances of the three definitional predicates: Non-specific Definitional (NSD) (any type of information relevant in a detailed definition of a term), Genus (category to which the term belongs) and Species (describes properties other than or in addition to genus). Machine learning and rule (lexico-syntatic) based approaches were used to extract predicates. The data-driven analysis uses techniques from summarization to cluster and order the entire set of NSD sentences based on properties of the data set as a whole. Then the definitional answer was generated by combining predicate information and data driven analysis or summarization result.

3.4 Non-Factoid Question Answering for Japanese

The work in [11] describes a system for answering non factoid Japanese questions by using answer type based weighting passage retrieval methods. They classified the non-factoid questions as definition-oriented, reason-oriented, method-oriented, degree-oriented, change-oriented, and detail-oriented questions and used a particular method for each category. The system comprises of prediction of type of answer, document retrieval, and answer extraction.

The system predicts the answer type of a question based on the interrogative phrase and extracts terms from the input question by using morphological analyzer. The score of every document is calculated and the top 300 documents with higher score or that are likely to contain the correct answer are gathered during the retrieval process to be used by the answer extractor. Then the answer extractor chunks the retrieved documents into paragraphs and retrieves those that contain terms from the input question and a clue expression. As a result, the system outputs the retrieved paragraphs as the preferred answer.

3.5 Singapore Definitional Question Answering

The work in [14] describes a system for answering definitional questions for Singapore. The system proceeds to construct definitions in three main steps: document and passage retrieval, sentence retrieval and sentence selection.

The document and passage retrieval component is to get relevant sentences about the search term. They employed a standard information retrieval system with anaphora resolution. The sentence retrieval module, which is the object of the study, integrates statistical ranking and pattern matching to produce a list of definition sentences. The final stage is sentence selection component to choose non-redundant definition sentences from the results of sentence retrieval to form the definition.

3.6 Arabic Non-Factoid Question Answering

The work in [13] presents a definitional QA system for the Arabic language called *DefArabicQA* that identifies and extracts the answers from Web resources (Arabic version of Google and Wikipedia) using rule-based approach. This QAS consists of four major components question analysis, passage retrieval, definition extraction, and ranking candidate definitions.

The first stage is question analysis where the question topics were identified by using lexical question patterns and the expected answer types were deduced from the interrogative pronoun of the question. The passage retrieval component collects the top-n snippets retrieved by the web search engine and only those snippets containing the integrate question topic are kept. Then the definition extraction component identifies candidate definitions from collected snippets by using lexical patterns under identifying candidate definition subcomponent and identified candidate definitions are filtered by using heuristic rules under filtering candidate definition subcomponent. After filtering the extracted candidate definitions, the extractor ranks them by using global score. Finally, the top five ranked candidate definitions are presented to the user. The performance of DefArabicQA is assessed by Arabic native speakers and mean reciprocal rank is also used as evaluation metric.

3.7 Portugese List Question Answering System

The work in [34] presents a web based List Question Answering System called LX-ListQA that focuses on answering portugese list questions where the answers are extracted and composed from several documents retrieved from the Web. The paper addresses problems that must be dealt with when answering list questions. This QAS consists of three main modules question processing, passage retrieval and answer extraction.

The first module is question processing, contains question analysis subcomponent which is responsible for cleaning the questions and expanding keywords using nominal and verbal expansion algorithm. The passage retrieval module searches web pages, clean and save their content information into local files, relevant sentences are selected based on matching and counting the keywords in the sentence. The researcher classified sentences into three classes according to their relevance with respect to the root question as weak, medium and strong. Finally, the answer extraction module identifies and extracts relevant answers and present them in the form of list. In this module two tasks are performed candidate answer identification which extracts all words tagged with the proper name and building the list answer based on frequency and rules.

3.8 English List Question Answering System

The work in [35] presents a QAS for answering English list question based on Distributional Hypothesis approach, which states that words occurring in the same

contexts tend to have similar meanings. The researcher used four components: answer type recognition, document retrieval, candidate answer extraction, and clustering.

First, the answer type of the question is determined through the answer type recognition component. Then, two different queries for searching the Web and the corpora are generated using the target and the question text. They use these two queries to create a collection of documents in which they look for the answers to the question.

The document retrieval component allows extracting from the collection all terms that comply with the answer type, which constitutes the initial candidate list. A similarity value is then computed for each pair of candidate answers based on their co-occurrence within sentences. Having clustered the candidates and determined the most likely cluster, the final candidate answers are selected.

3.9 Summary

Research has been done on Afaan Oromo factoid type questions (person, place, number and time) where the answer is entity name extracted by rules and gazetters from documents. Afaan Oromo List, definition and description questions are non-factoid type question where the answer is complex than the factoid question which requires explanation. Therefore, it is impossible to use the factoid QAS for answering definition and description question. But, list question is an extended version of factoid question which retrieves answers from different documents.

There are works on foreign and local languages for answering definition and description question which cannot be used for answering Afaan Oromo questions because of the morphological and grammatical structure difference. However, the QAS developmental techniques of those local and foreign languages are adopted to develop an Afaan Oromo list, definition and description QAS.

Chapter Four: System Design and Implementation

In this chapter the architectural design and detailed implementation of AOLDDQAS will be presented. The first section describes the main components, integration and architecture of the proposed system. The document pre-processing section presents the architecture of AO analyzer and explains the techniques. The third section covers detailed strategies and algorithms implemented in analyzing and generating questions. The next section covers the specific methods used in retrieving and filtering documents retrieved from the corpora. The fifth section details about the techniques and algorithms used for selecting the best answers. Finally the summary section summarizes the chapter.

4.1 Architecture of AOLDDQAS

In this thesis, we propose a type-by-type approach for answer extraction. The design and implementation process of the proposed QA system consists of four major phases which are document pre-processing, question analysis, document selection, and answer extraction. The document pre-processing module preprocesses documents. Question analysis determines question types, pre-process queries and constructs proper query for the document selection component. It includes question classification and query generation subcomponents. The document selection component is responsible for retrieving documents using the generated query by document retrieval subcomponent and filtering using document analysis which will be presented to the answer extraction. The final component, answer extraction, is responsible in extracting and presenting answers. The Definition-Description Answer Extraction and List Answer Extraction were used. Figure 4.1 shows the general architecture of Afaan Oromo List, Definition, Description Question Answering System. Except the morphological analyzer used for the lemmatization, the AOAnalyzer component that we used is similar to that used in [22].

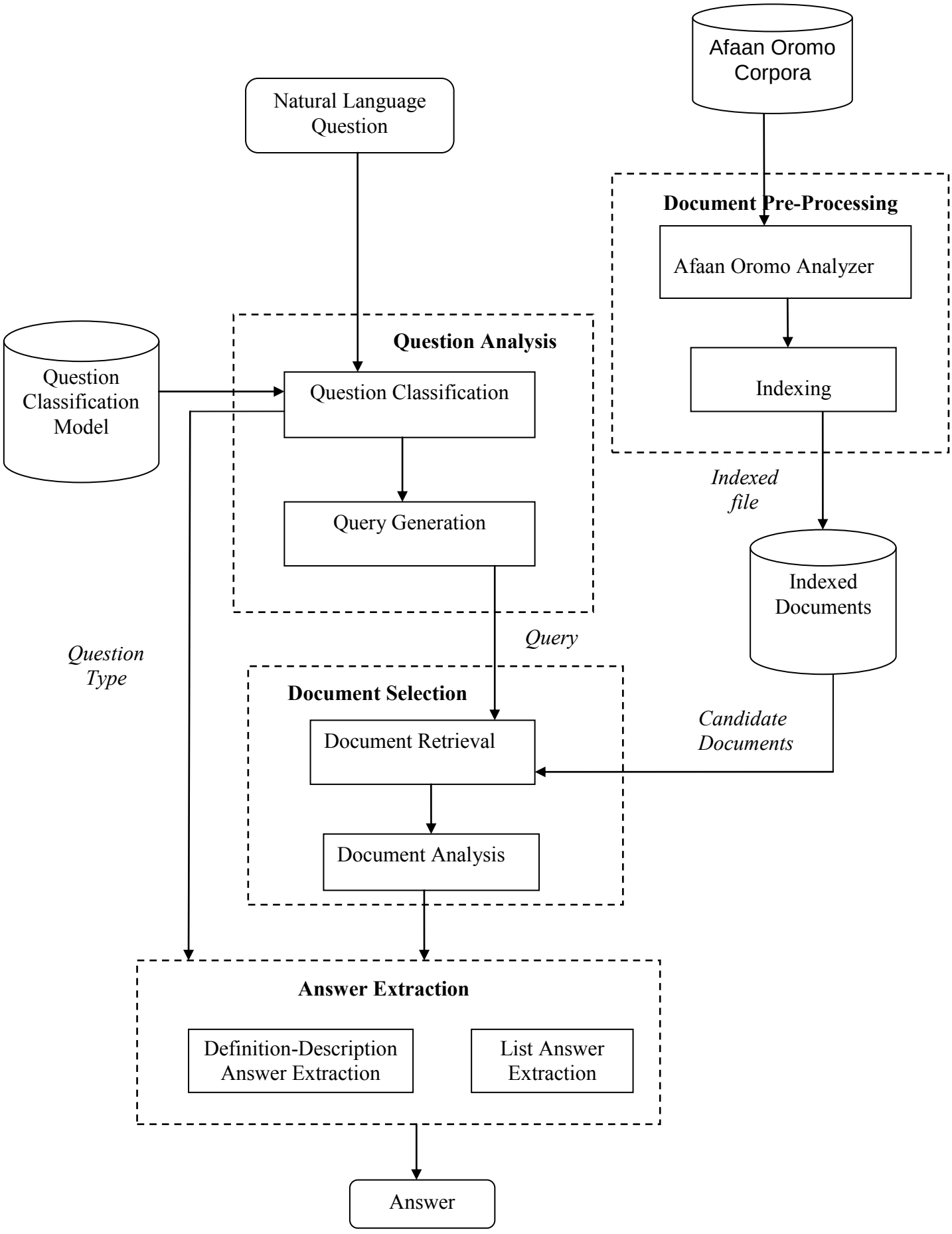

Figure 4.1: *Architecture of AOLDDQAS*

4.2 Document Pre-processing

The documents for this thesis work are Afaan Oromo corpora collected from different web sites and books. In the process of question answering activity, before retrieving documents which contain an answer of a question from Afaan Oromo corpora, different text pre-processing tasks are involved. The main pre-processing techniques we have used are text tokenization, short word expansion, case normalization, stop word removal, stemming, morphological analysis and indexing that should be done in order to accomplish the question answering task. Figure 4.2 shows Afaan Oromo Analyzer used for pre-processing questions and documents.

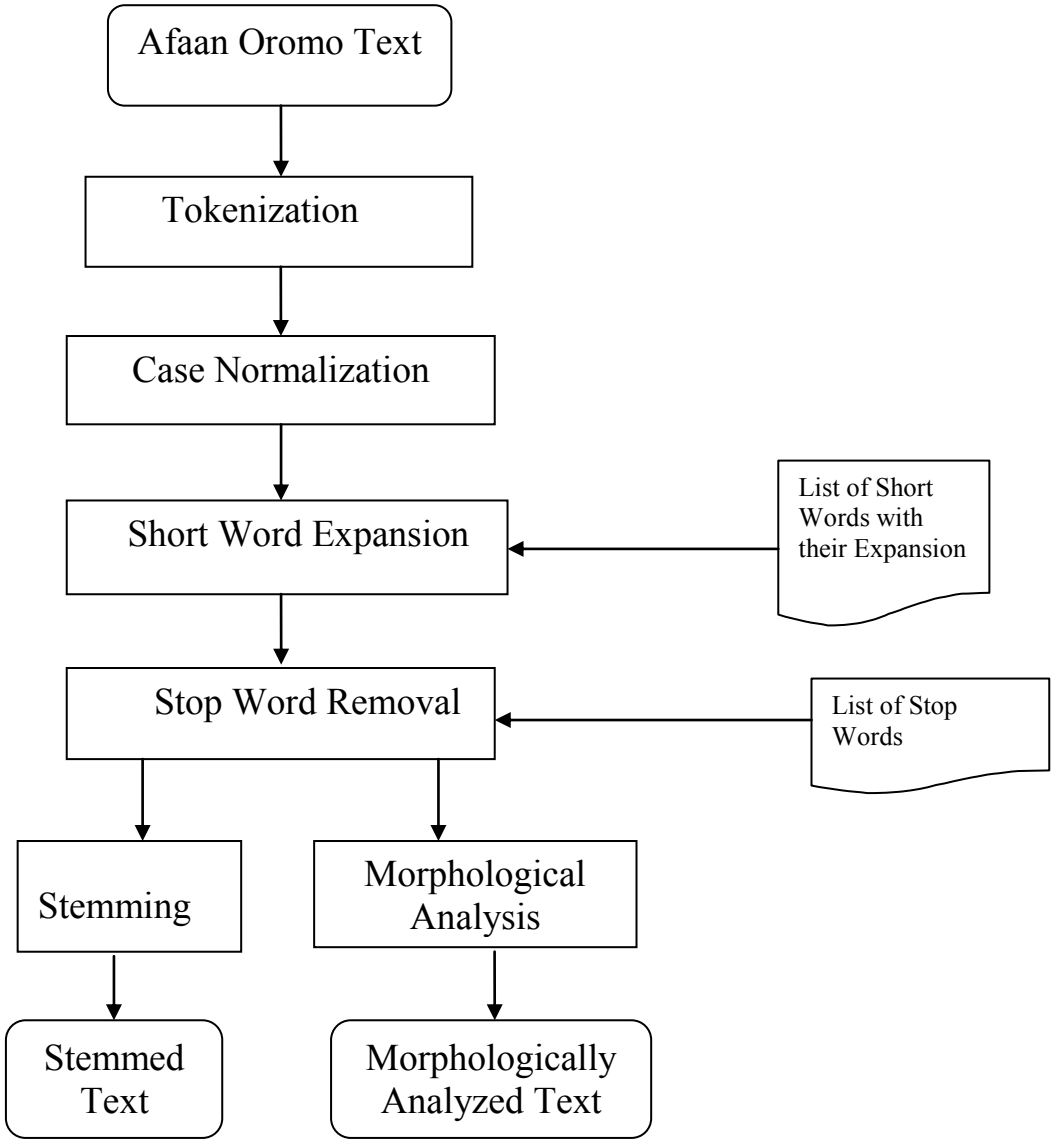

Figure 4.2: *Architecture of Afaan Oromo Analyzer Component*

i) Tokenization

Tokenization breaks the stream of characters into raw terms or tokens, detects word boundaries of a written text and at the same time it can be taken as the process of removing non alphanumerical characters. The document retrieval subcomponent of question answering system fetches answers of natural language questions from Afaan Oromo indexed files. Tokenization is one of the text pre-processing tasks that should be done before the file indexing task.

Tokenization helps in matching the tokens of the query with tokens in the document. In Afaan Oromo white spaces are used to separate the boundary of a token and punctuation marks such as commas, periods, question marks, exclamation marks and hyphens are important to demarcate the boundaries of tokens but, in Afaan Oromo language apostrophe mark is considered as a part of a word. For example, in the word “sa'a” (cow), the apostrophe is used to show that the vowels are produced independently. Thus, the word “sa'a” has to be treated as a single token in the tokenization process. White spaces and punctuation marks except ““”, “/”, “.” “,” and “-” are used as a word delimiters. For example, if the original document is "*Oromiyaan, qabeenya uumamaan badhaatuu dha*" the tokens will be '*Oromiyaan*', '*qabeenya*', '*uumamaan*', '*badhaatuu*', '*dha'*. Thus, we can define *token* as an instance of a sequence of characters.

ii) Case Normalization

Normalization can be defined as the process of transforming text into some other forms. It is the process of handling problems related with variation of cases (UPPER CASE, or lower case or Mixed Cases). So the good way to handle this problem is converting the whole document into similar case. In some languages like Amharic which does not have a distinction between upper and lower case, this might not be a big deal. However, it is very important for languages that use Latin characters for writing. In this research we will use lower case letters for questions and corpus.

iii) Short Word Expansion

Documents could contain words written in short forms. Short words are short form of words or phrases which can be formed from initial letters of important terms of a word or a phrase or from the combination of letters of a word or a phrase and other characters. Usually in Afaan Oromoo, '.' and'/' are used while writing words in short form. For

example, if someone asks the question "M/B maali" (What is school?) the word “M/B” should be expanded to “mana barumsaa” (school) while searching the answer to the question. Some short words and their expansions are shown in Appendix 1.

iv) Stop Word Removal

Stop-words are most frequent terms which are common to every document, and have no discriminating power. Thus, these terms should not be considered in *indexing* process. List of some stop words are shown in Appendix 2.

v) Stemming

Stemming is an activity to find the stem of a word by removing affixes, i.e., it enables to merge morphological variants of a word under a single index entry or its common form. Thus, for this research work, we have used Debela‟s stemmer, which takes a word as an input and removes its affixes using a rule based algorithm [15].

vi) Morphological Analysis

Morphological analysis is the segmentation of words into their component morphemes and the assignment of grammatical morphemes to grammatical categories and lexical morphemes to lexemes [25]. Thus, the morphological analyzer returns the root of a word and it enables to merge morphological variants of a word under a single index entry or its common form.

HornMorpho[33] is used for the morphological analysis task. HornMorpho is a Python program that analyzes Amharic, Oromo, and Tigrinya words into their constituent morphemes (meaningful parts) and generates words, given a root or stem and a representation of the words grammatical structure [25].

vii) Indexing

Indexing is a process that converts documents in a repository into cross reference lookup (index). The index stores statistics about terms in order to make term-based search more efficient. It is the last step on document preprocessing, i.e., before being indexed, it is necessary to perform the techniques discussed above.

For this purpose, a Lucene [32] library which categorized in inverted index family was used. Lucene is a high performance, scalable information retrieval library. It has facilities

for text indexing and searching that can be integrated into applications. Lucene index contains a sequence of documents, where a document is a sequence of fields and a field is a named sequence of terms.

4.3 Question Analysis

The main function of the question analysis component is to understand the kind of information the question is asking for. In addition, it is responsible to formulate proper queries for document retrieval. When the user poses a question to the system, the question analysis component takes in the user query and passes it to its sub components.

4.3.1 Question Classification

For answer extraction in a large collection of documents and texts, at first the system should know what it looks for. In this case, questions should be classified regarding their types. The question classifier subcomponent determines the type of a question as list, definition or description.

Table 4.1: *Some Question classes, interrogative terms, and class indicative terms*

Class	Interrogative term	Question type Indicator term
Definition	Maali, maalinni, maalidha, maal jechuu dha, jechuun maal jechuu dha, maalisheen, maalisaanni	Hiikni, Hiikni [isaa|ishee|ishii|isii|isee|isaanii], hiikaan [isaa|ishee|ishii|isii|isee|isaanii] ,hiiki [isaa|ishii|ishee|isii|isee|isaanii, yeroo[hiikamu| jennuu|hiiknu]
Description	maali, maalidha, maaliif, akkami keena|kenitti|kennu|qaba|qabdi|qabu, oola|oolu, akkamirra|akkamiif [oola|oolu]	faayidaa,gayee|gahee|ga'ee [isaa|isaani|ishi|isii|isee|ishee], faayidaan[isii|isaa|ishii|ishee|isee|isaani], dalagaa[wwan|isaa|ishee|isaanii], fayyada|fayyadi|fayyadu
List	ibsi|tarreessi|tarreessa|ibsaa|caqasii |caqasaa, maal fa'i, maal fa'aa dha, barreessi|barreessaa|eenyu fa'i|eenyuu fa'a dha.	Ulaagaa[lee|wwan],sababa|sababoota, madda|maddoota,seera|seeroota,mala|maloota, gosa|gosoota|qoqqoodama,hariiroo[wwan], kaayyoo[wwan], harroo[wwan],naannoo[lee], magaala[oota],dammee[wwan],mirga|diirqama ,sababata'an, maal akka ta'ee, akaakuu|akaakuuwwan

Question classification is a subcomponent of question analysis which is concerned with assigning questions to semantic classes. This semantic classification can be used to reduce the search space of possible answers. In order to assign a question to semantic classes, a rule based question classification and machine learning approaches are used most of the time but, for this thesis we only used a rule based approach to determine the type of question. This is due to the study in [18] and [22] which show that the rule based question classification approach got good performance than the machine learning approach. Thus, we have used a rule based approach to identify type of questions using Algorithm 4.1. The algorithm determines the question type by using the interrogative terms of the question and class indicative terms shown in Table 4.1 For example, given the question "Faayidaan ekistenshinii fayaa maali?" ("what is the use of health extension?") the terms "faayidaa" (use) and "maali" (what) indicate that the question is looking for a description. Another example for definition question, "hiikni Gadaa maali?" ("what is the meaning of Gedaa?") the terms "hiikni" (meaninig) and "maali" (what) indicate that the question is looking for definition question. Question asking for a thing "Diirqama barataa caqasii?" ("list students duty?") the terms "caqasii" (list) and "diirqama" (duty) indicates that the question is looking for list question. Question asking for a place "harroowwan umamaa Itoophiyaa tarreessi" ("name Ethiopian natural lakes") the terms "tarreessi" (name) and "harroowwan" (lakes) indicates that the question is looking for list question. Algorithm 4.1 shows a rule based question classification method for classifying queries to their classes.

```
Input the question
If the question contains (one of the definition indicative
terms) then
Return question type "Definition"
Else If the question contains (one of the definition
question particles and  (one of the definition indicative
terms) then
 Return question type "Definition"
Else If the question contains (one of the description
```

```
indicative term and one of the descriptive question
particle) then
Return question type "Description"
Else If the question contains (one of the list indicative
term and one of the list question particle) then
Return question type "List"
Else
     Return question type "Unknown"
 End If
```

Algorithm 4.1: *Rule Based Question Classification Algorithm*

4.3.2 Query Generation

Query generation is used to convert the users‟ natural language questions into suitable form for document retrieval. First users' query will be pre-processed using AOAnalyzer on Figure 4.2 which contains tokenization, case normalization, stop word removal, short word expansion, stemmer and morphological analyzer tasks. Then, the query generator removes interrogative terms from the pre-processed query and generates a query which is used by the document retrieval. The question particles like "maali"**,** "jechuun maal jechuu dha", "faayidaan", ulaagaa, tarreessi, caqasii etc., are removed from the question because it doesn‟t worth for searching. Algorithm 4.2 is used for generating queries and the interrogative terms are listed in Table 4.1. Finally, the generated query is sent to the document retrieval component.

```
Input the query
Pre-process query
If(question contains interrogative terms) then
Remove interrogative terms
Return generated query
End if
```

Algorithm 4.2: *Query Generation Algorithm*

4.4 Document Selection

The document selection component consists of the document retrieval sub component which is responsible for retrieving documents which may contain information pertinent to the list, definition or description of a target and document analysis subcomponent responsible for filtering documents by identifying the relevant document from the irrelevant one.

4.4.1 Document Retrieval

Document retrieval has a responsibility to fetch documents which are related to the generated query, it takes keywords produced by the query generator component. It starts with user‟s query and terminates with a list of documents ready to be processed by document analysis and later for answer extraction, also used as an intermediary between question analysis and answer extraction components. For this study, a Lucene package [33] was used for searching. It returns a ranked list of candidate documents by considering the number of keywords of the query in the documents from the Lucene index. As a result, the set of related documents to the given query will be returned as Hits.

4.4.2 Document Analysis

Retrieved documents should be filtered before further analysis in order to identify relevant documents from the irrelevant. Thus, document analysis first locates the question keyword using keyword extractor [27] and based on the keyword it filters the documents.

Keyword Extractor (KE) extracts the keyword/target term(s) of the question [4]. Keyword/target term is obtained by removing indicator terms from the query, the query term is used later in sentence extraction subcomponent.
After the target is extracted, the documents will be tested by their respective rules (regular expressions) listed for (definition, description, and list where, the focus is thing) in Table 4.2. Then, if a text is extracted by one of the rules, the document will be kept; otherwise it will be removed. The list question with the place focus is tested by gazetter.

4.5 Answer Extraction

Answer extraction is the final stage and most important in a QA system. The goal of answer extraction is to extract and select answer from a pool of answer candidates and present the most likely answers for a question. This research used a type by type answer extraction method as Definition-Description Answer Extraction (DDAE) and List Answer Extraction (LAE). Figure 4.3 shows Definition-Description and List Answer Extraction components.

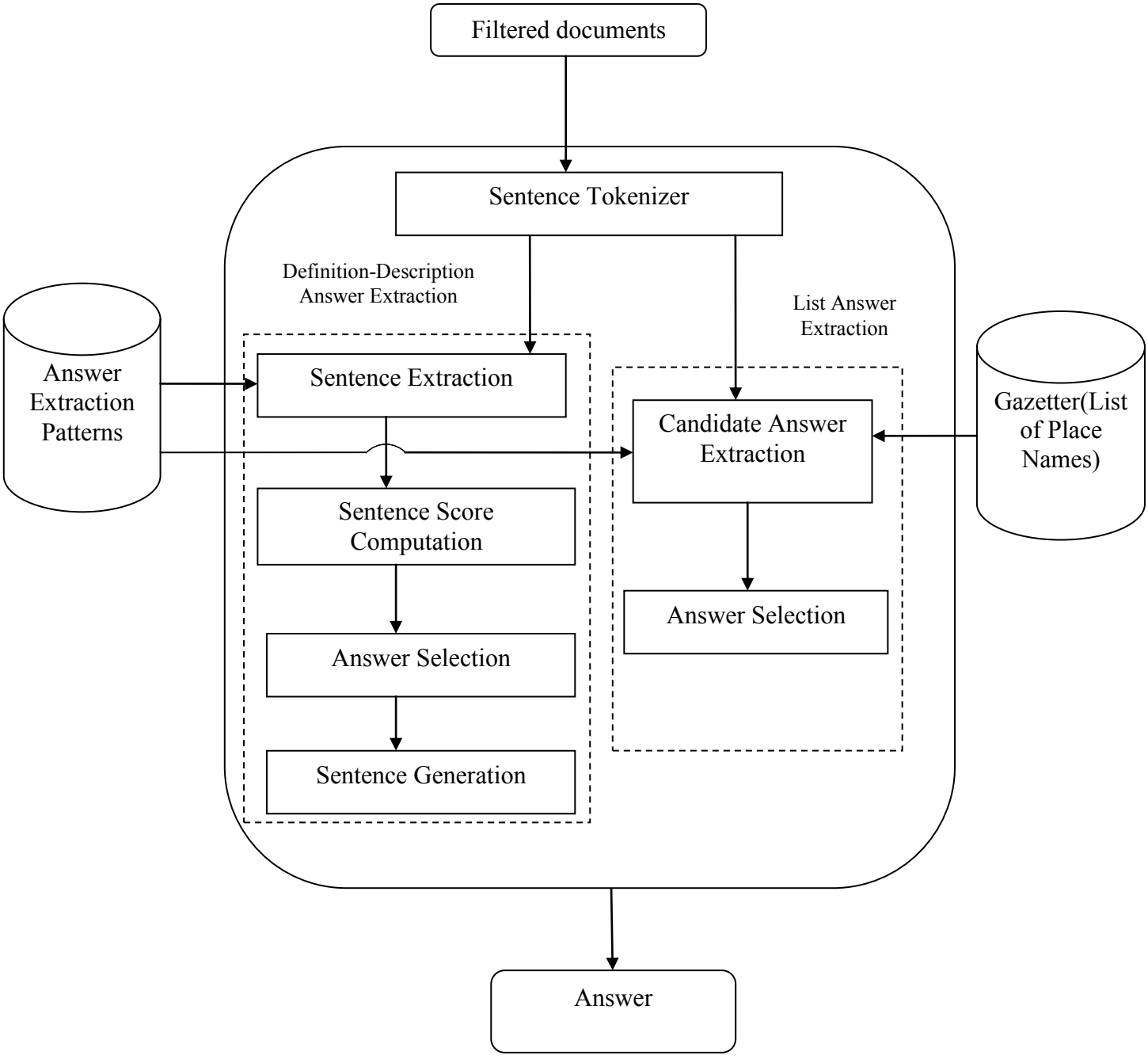

Figure 4.3: *Definition-Description and List Answer Extraction component*

Sentence Splitter/Tokenizer

In order to achieve the goal of answering questions, first the filtered documents have to be tokenized. Sentence tokenizer is the first component in answer extraction, used to split/tokenize the filtered documents retrieved from document analysis component using (".", "!","?") characters as delimiter and pass the tokenized sentences to the DDAE and LAE subcomponents for further processing.

4.5.1 Definition-Description Answer Extraction

The DDAE component used a pattern matching method for extracting answers. Definition and description (non-factoid) questions require a short paragraph which briefly defines the target or state concepts the user wishes to know more about unlike the factoid question which requires a single fact [31]. Thus, finding snippets or piece of information about the current target, ranking, selecting, and generating them is very important. To do so, the answer extraction consists of snippet/sentence extraction, sentence score computation, answer selection, and sentence generation units.

Sentence/Snippet Extraction

Sentence extraction is a technique used for automatic summarization of a text [27], works as a filter which allows only important sentences to pass and creates candidate answer set. There are manually crafted indicative patterns that are listed in Table 4.2: allows sentence extractor to extract sentences from the tokenized sentences using Algorithm 4.3.

Table 4.2: *Sentence/Nugget Extraction Patterns*

Question Type	Sentence Extraction Patterns
Definition	Rule 1:target+"jechuun"+".*" Rule 2:target+"jechuun"+".*"+"jechuu dha" Rule 3:target +" .*"+ hiikni|hiiki|hiikaan[isaa] +".*"+jechuu dha" Rule 4:".*"+ target+" .*"+"(hiikamuu|waamamuu)"+"danda'a" Rule 5:target + " .*"+"(yeroo[ta'u|jedhamu|waamamu])"+" .*" Rule 6:".* " + target +" jeedham[a|u|ti]" Rule 7: target+".*"+"jeedhamuun"+"(hiikama|beekama| waamama)" Rule 8:target +" .*"+"(dha|argama|qaba|qo'ata|ibsuu dha)" Rule 9:target+" .* "+"jechuun ni danda'ama|jechuu dha" Rule 10: ".* "+target+"jechuun"+".* " Rule 11:".*"+"target +".*"+ "jechuun"+".*"
Description	Rule 1:target+".*"+"(faayidaa[lee]|gahee |ga'ee | hiree[wwan|n])"+".*" + "(kan qabu|kan qaban|kan qabdu)"+ ".*"+"(ni [oola|oolu]| ool[a|u|ti] |qabu |qaba |qabdi |ni[qaba|qabu] | kenna | kennu| kennitti|dha)" Rule2:target+".*"+"hojiiwwan[fayyada|gargaara|oola]|(fayyadu|gargaaru|olu[danda'a|danda'u|dandayaa|dandeeyssi] Rule 3:".*"+faayidaa[n|wwan]+target Rule 4:target+".*"+"(fayyada|gargaara|qarqaara|fayyadu|gargaaru|qarqaaru |ni[fayyada|gargaara|fayyadu|nigargaaru]|waanfayyaduuf|waangargaaruuf| waan qarqaaruuf)" Rule 5:target+".*"+"(kan[fayyadu|fayyaddu|fayyadan|gargaaru|gargaaran| dandesiisu|dandeessisani)"+".*"+"dha" Rule 6:target+".*"+"gargaara|fayyada"+".*"

List	Rule 1:target+".*"+(isaanis|kan|dalagaan isaas|kanneen keessaa muraasni) Rule 2:target+".*"+(bakka|akka kannatti)+".*"+qoodama|qoodamu. Rule 3:dalagaa(n|wwan)+target Rule4:maddi|dirqama|mirga|karaalee|(kaayyoo|dammee|hariiroo(n|wwan)) +target Rule 5:hariiroo|kaayyoo(wwan)+target+beekamoo ta'an Rule6:".*"+dalagaa(n|wwan)+target Rule 7:dammee(n|wwan)+target+muraasni Rule 8:".*"+target+".*"+jeedhamu|jiru(isaanis) Rule 9:karaalee+target+ittisuuf fayyadan Rule 10:kannen armaan gadii+target+beekamoo dha Rule 11:target+bakka+".*"+qoodamu ni danda'uu isaanis Rule 12:".*"+target

```
Input filtered documents
For all filtered documents
Split sentences //sentences are splitted using
(".","!","?") as delimiter.
Flag =0 // indicates if the sentence matches the pattern
or not.
If (splitted sentence matches patterns) then
Flag=1
Return sentence
Else
Discard
End If
End For
```

Algorithm 4.3: *Sentence Extraction Algorithm*

Sentence Score Computation

An answer to a question should contain all the vital snippets or sentences. Thus, in order to select the appropriate sentences from the candidate answer set, we used the sentence scoring function given in Equation 1 from [18], i.e., the score of a sentence S is calculated as the sum of the percentage of the query (Q) terms in the sentence, weight of the pattern that identifies the sentence, the reciprocal of the position of the sentence in the document that contains it, and the Lucene score of the document D that contains S.

$$\text{score}(S) = \frac{N_{S \cap Q}}{N_Q} + weight\ (S,\text{P}) + \frac{1}{pos\ (S)} + luceneScore\ (D,S) \quad (1)$$

Where, $N_{S \cap Q}$ is the number of terms that are found in both S and Q, N_Q is the number terms in Q,$weight\ (S,P)$ is the weight of the pattern P that matches with S, $luceneScore(D,S)$ is the score of document D that contains S by Lucene, and $pos(S)$ is the position of S in the document that contains S.

Since the position of a sentence does not have any impact for description questions, score of sentence S is computed by the formula given in Equation 2.

$$score(S) = \frac{N_{S \cap Q}}{N_Q} + weight\ (S,\text{P}) + luceneScore(D,\text{S}) \quad (2)$$

Answer Selection

During sentence extraction, sentences are extracted from the sentence tokenizer subcomponent then, we need to determine if any two sentences contain roughly the same information. That is, given sentence A does sentence B provide any new and relevant information for the purpose of defining a given target [27]. As the work in [27] suggested one way of determining the similarity of texts is to use word overlap (if sentence B contains novel information about the target when compared to sentence A). The more different text fragments share common non-stop words, it indicates that they are highly similar [31]. A formula is adopted from [27] which calculate the similarity between sentence A and B but first a sentence profile is constructed for each sentence which contains the set of non-stop words, T, in the sentence. This is the percentage of tokens in A which appear in B or the percentage of tokens in B which appear in A.

$$\text{sim}(\text{A}, \text{B}) = \frac{|\text{T}_\text{A} \cap \text{T}_\text{B}|}{\min(|\text{T}_\text{A}|, |\text{T}_\text{B}|)} \quad (3)$$

Where, $sim\ (A,B)$ is the similarity of the sentences A and B, $|T_A|\ and\ |\ T_B|$are the number of non-stop tokens in sentences A and B respectively, and $|T_A \cap T_B|$ is the number of common tokens in A and B.

Thus, in order to construct the final answer the answer selection subcomponent ranks the sentences by their score, selects the top ranked sentences according to the length requirement, and avoids introducing any redundant sentences into the result. We have used Algorithm 4.4 for answer selection from [27].

```
Input set of sentences in the candidate answer set with
their respective score
Sort all the sentences in the candidate answer set using
their score
For all sentences in the candidate answer set
Add sentence in the pool
Examine the similarity of the next sentence S with
sentences in the pool
If(similarity of the sentence S is greater than or equal to
0.7)
Skip sentence S
Else
Add sentence S to the pool
End If
End For
Return Top 6 non-redundant sentences.
```

Algorithm 4.4: *Answer Selection Algorithm*

Sentence Generation

Finally, this subcomponent integrates sentences which passes through the above subcomponents and display the result to the user. Using the Algorithm 4.5 sentence generation will be done. Thus, it generates the sentences in a way that, sentences that begin with the target term will be positioned at the beginning, sentences that begin with connective terms which are listed in Table 4.3 will be in the middle, and sentences which

start with other terms will be after the others. The sentences score is used for ordering sentences that have the same priority.

Table 4.3: *Some of Afaan Oromo Connective Terms*

Waan taeef	kanaafu
Ega ta'eef	akkasumas
Waan ta'eefu	kanarran kan ka'e
Kanaaf	garuu
Kana	ta'uus
Kunis	ijaa ta'eef

```
Input candidate sentences
For all candidates
If(candidate sentence starts with the target term)then
Put the sentence at first
Else If (candidate sentence starts with connective terms) then
Put the sentence at the middle
Else If (candidate sentence starts with other terms)
Put the sentence next to the middle sentences
Return ordered sentence
End If
End For
```

Algorithm 4.5: *Sentence Generation Algorithm*

4.5.2 List Answer Extraction

Answer extraction is selection of an answer for a given query from collection of text documents. Answering List questions is more difficult compared to answering factoid questions because it requires a system to acquire the answer instances from different sources (answer fusion). This component is used to answer two types of list question. Before extracting candidate answers, the filtered documents are tokenized. Depending on the question focus LAE used two methods for extracting candidate answer from the tokenized sentences. The first one is for answering about things in this case, the pattern matching method in Table 4.2 were used for extracting the tokenized sentences. On the

other hand, if the question focus is places, gazetter based named entity recognition were used. The gazetter of place names includes regions of Ethiopia, some of Oromia cities, lakes and rivers of Ethiopia. The LAE has two subcomponents candidate answer extraction and answer selection. Candidate answers are extracted from the tokenized candidate sentences which are returned by the document filtering with the help of the written regular expressions for things extraction and gazetteers for the extraction of place names. Next, answer selection will select the best answer from the pool of candidate answers. Candidate answers which have higher weight (more number of query terms) and repeated in more than one sentence will be considered as correct answers. Algorithm 4.6 is used to extract answers to list questions.

```
Accept filtered documents returned by document analysis
Tokenize documents into sentences using (".","?","!")
characters as delimiter
For all tokenized sentences
If question focus is thing
Extract the candidate answer using pattern matching
Else if question focus is place
Extract the candidate answers using list of gazetters
End if
End for
For all extracted candidate answers
Calculate similarity
Select top answers
End for
```

Algorithm 4.6: *Algorithm for List answer extraction component*

4.6 Summary

This chapter described the architectural design of the AOLDDQA system and the implementation of its main components. The AOLDDQA system implementation consists of four main modules. The document pre-processing component is used to normalize, remove stopwords, expand short words, stem, lemmatize and index documents. Once documents are normalized and indexed, they will be ready for the

succeeding components for further processing. Question pre-processing is used to manipulate the questions to create a proper query and is done in query generation.

The question analysis component determines question type by using a rule based technique and pre-process and creates a proper query that will be submitted to the document selection component. The document retrieval component retrieves documents using the query from the question analysis component and filters the retrieved documents using filtering patterns. The answer extraction component has sentence tokenizer, DDAE and LAE. DDAE contains sentence extraction subcomponent which extracts sentences from the sentence splitter using manually crafted answer extraction patterns. The score of each sentence is computed by the sentence scoring subcomponent. Then, the answer selection algorithm selects top 6 non-redundant sentences from the candidate answer set. Finally, the sentences are generated by the sentence generator subcomponent and returned to the user. LAE contains candidate answer selection, rules and the gazetteers are incorporated to extract answers. Questions asking about a thing are matched with the rules developed and place name based questions are matched with the gazetteers and answer selection selects the answer.

Chapter Five: Experiment

The focus of this chapter is on evaluating our system. The first section describes the methods and systems used in creating the prototype. The next section details the evaluation criteria for question classification a percentage and precision, recall and F-score for document selection and answer extraction with the result obtained. The last section discusses the issues faced in doing the research.

5.1 The Prototype

We have prepared a prototype which can basically take user‟s natural language query and after going through all the mentioned processes, deliver an answer to the user. The algorithms we have developed are implemented using the Java programming language with eclipse java editor, document indexing and searching has been done using Lucene API. The system is developed and tested on a system with Intel® Core™ i5-3230 CPU of 2.60GHz, a 6GB RAM, a 600GB Hard Disk and a windows 8 operating system.

5.2 Evaluation Criteria

Evaluation for QA system mainly focuses on the accuracy of the answers returned, i.e., correctness of answers. It is done by comparing the answers returned by the system with human-proposed answers to the same set of questions. The accuracy of question classification module is done by evaluating the percentage of correctly identified question types. The percentage is computed by taking the ratio of correctly identified questions to the total test questions. The document selection and answer extraction modules are evaluated by precision, recall and F-score.

5.2.1 Question Classification Evaluation

Question classification is one task of question analysis used to classify questions into its intended types and identify an expected answer types. The performance of question classification is crucial for answer extraction, i.e., wrongly classified questions will lead to return wrong or No answer as result. The question classification is evaluated by the percentage of correctly and wrongly classified questions. Where, the percentage is computed by taking the ratio of correctly identified questions to the total test questions. The experiment is conducted on 300 test questions that are chosen from list, definition

and description question types and our system correctly classified 98%, 99% and 97% respectively. Sample questions and classification results are attached in Appendix 4.

5.2.2 Document Selection Evaluation

The standard approach to information retrieval system revolves around the concept of relevant and non-relevant documents [24]. Document retrieval systems are evaluated with respect to the notion of relevance judgment by human that a document is relevant to a query based on the presence of correct answer particles on the retrieved documents.

In information retrieval system, precision and recall are defined in terms of a set of retrieved and relevant documents as follows:
Precision is the ratio of the number of relevant documents returned to the number of documents returned, used to assess the measure of how many of the documents returned for a given query are actually relevant.

$$\text{Precision(P)} = \frac{\text{Number of Relevant Documents Retrieved}}{\text{Total Number of Documents Retrieved}} \quad (4)$$

Recall is the ratio of the number of relevant documents returned to the total number of relevant documents in the collection.

$$\text{Recall(R)} = \frac{\text{Number of Relevant Documents Retrieved}}{\text{Total Number of Relevant Documents}} \quad (5)$$

In order to get a synthetic measure from both precision and recall, the harmonic mean between the two (known as F1 or F score) is also used.

$$\text{F} - \text{score} = \frac{2*\text{P}*\text{R}}{\text{P}+\text{R}} \quad (6)$$

The document retrieval performance has been evaluated based on the presence of correct answer particles on the retrieved documents. The performance of our document selection component is evaluated by 75 questions on 250 documents. According to the query search result, our system scored a recall of 0.87, 0.686 precision and 0.767 of F-score.

5.2.3 Answer Extraction Evaluation

The answer extraction component is responsible to extract answer from the relevant documents which are retrieved by the document retrieval. It is evaluated using precision,

recall and F-score by comparing the answers that our system returned with manually constructed answers using 50 test questions for each question types. Table 5.1 shows evaluation result of answer extraction component. Figure 5.1-5.4 show examples of three classes that are correct answer, wrong answer (i.e., answers that contain unrelated concepts with the question), and no answer respectively.

Precision: percentage of instances returned that are correct

$$\text{Precision(P)} = \frac{\text{Number of correct answers}}{\text{Total Number of returned answers}} \quad (7)$$

Recall: percentage of the expected correct instances that are returned

$$\text{Recall(R)} = \frac{\text{Number of correct answers}}{\text{Total Number of expected answers}} \quad (8)$$

Recall: is a weighted harmonic mean of precision and recall, equation 6 were used to evaluate the quality of an answer.

Table 5.1: *The Answer Extraction component Recall, Precision, and F-score result*

Question Type	Precision	Recall	F-score
Definition	0.628	0.743	0.681
Description	0.561	0.719	0.63
List	0.6	0.706	0.648
Average	0.596	0.723	0.653

From Table 5.1, we conclude that the answer extraction patterns of definition are better than description because the F-score of definition is greater than the F-score of description. Description terms are incorporated within their definition which leads the F-score result of the description to be less. The F-score of list question is also good (better than the description). Figure 5.1(a), 5.1(b) and 5.1(c) are screenshots of correct answers for list, definition and description questions.

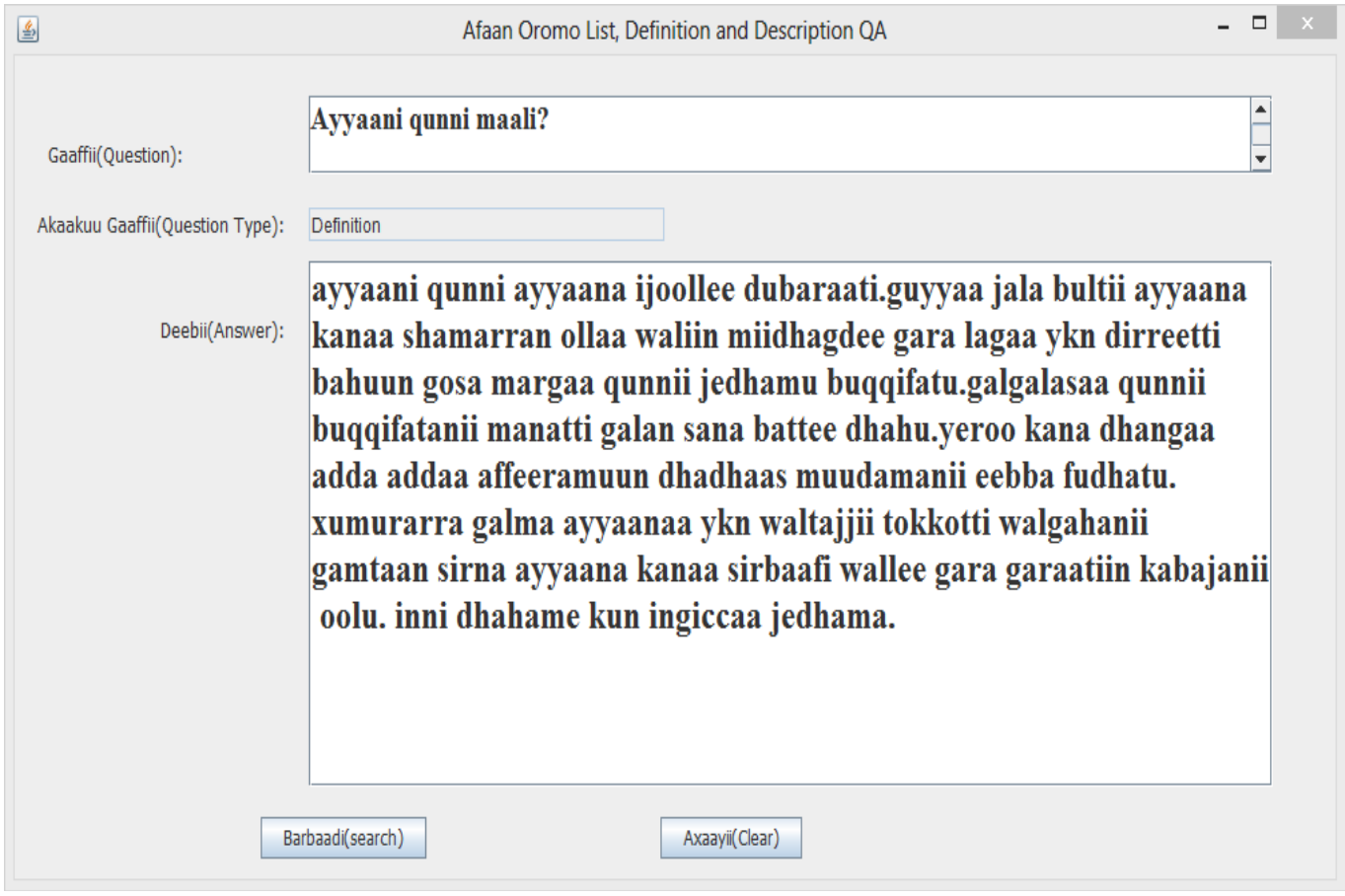

Figure 5.1 (a): *Screenshot of Correct Definition Answer Example*

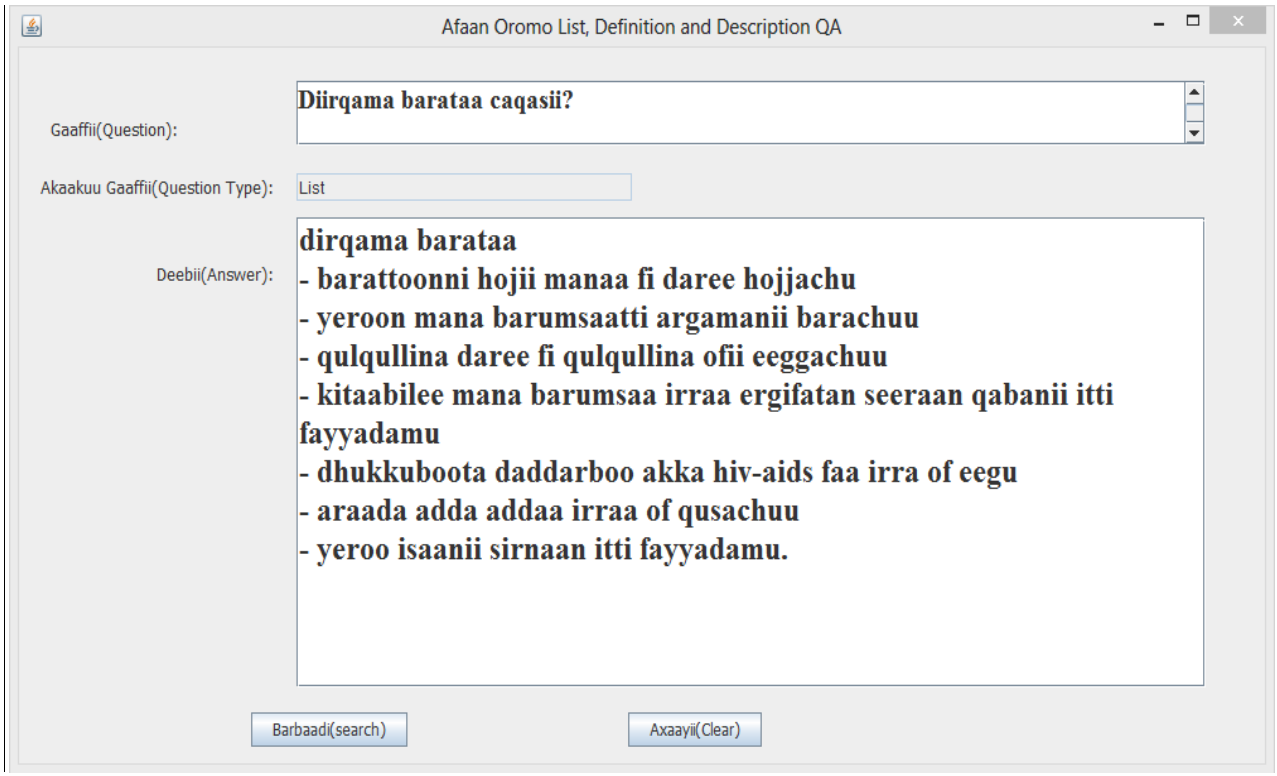

Figure 5.1 (b): *Screenshot of Correct List Answer Example*

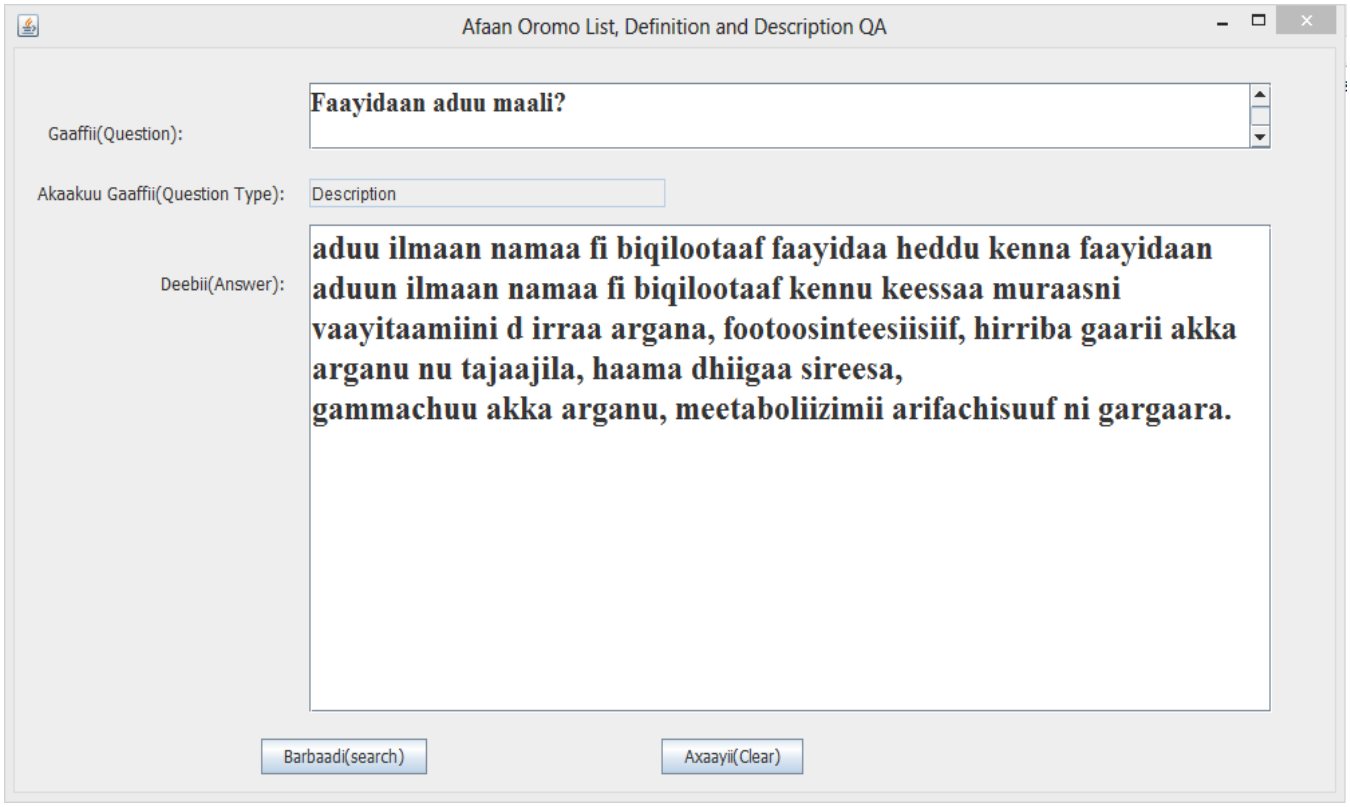

Figure 5.1 (c): *Screenshot of Correct Description Answer Example*

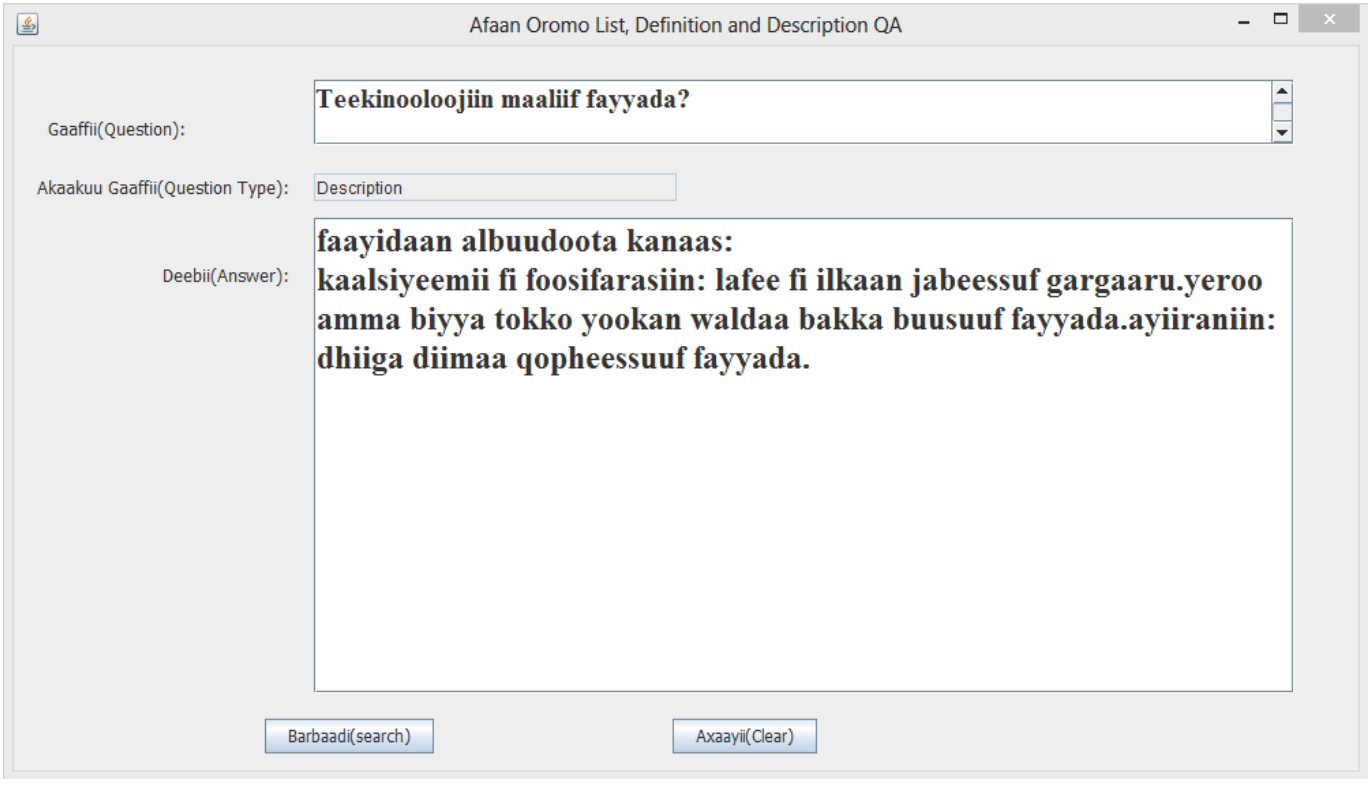

Figure 5.2: *Screenshot of Wrong Answer Example*

As Figure 5.2 shows the system returned wrong answer for the question "teeknoolojiin maaliif fayyada?" ("what is the use of technology?")"faayidaan albuudoota kanaas: kaalsiyeemii fi foosifarasiin: lafee fi ilkaan jabeessuf gargaaru. yeroo amma biyya tokko yookan waldaa bakka buusuuf fayyada. Ayiiraniin dhiiga diimaa qopheesuuf fayyada". The answer has no relation with the question asked, the system used irrelevant document.

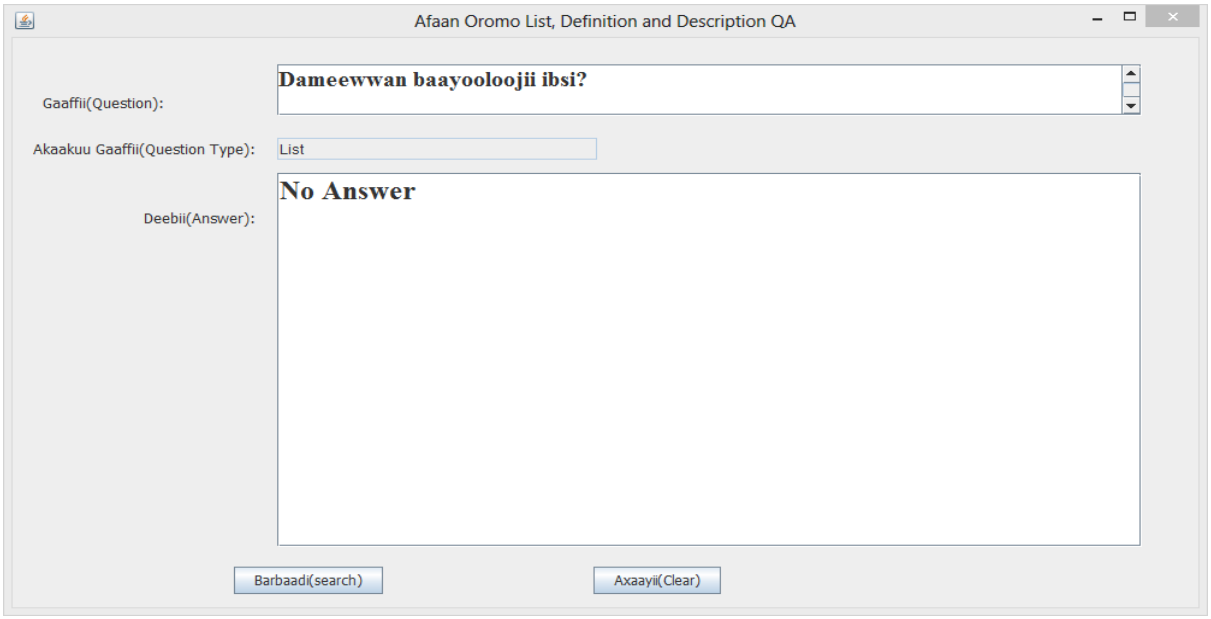

Figure 5.3: *Screenshot of No Answer Example*

As Figure 5.3 shows, the answer for the question "dameewwan Baayooloojii tarreessi?" ("list branches of Biology?") return No Answer but, there is a document in the corpus about Biology. The error occurred due to document retrieval.

5.3 Discussion

During the evaluation we used two evaluation criteria. The first evaluation criterion is used for question classification component, computes correctly classified question types. The other evaluation criterion was precision, recall and F-score used for evaluating document selection and answer extraction components. In doing so, we faced some issues which are listed below.

- Spelling errors in extracting correct answer, for example instead of writing the word "aadaa" (culture) if the question term is written as "adaa" it leads to return no answer.

- Even though the techniques we used in this thesis have performed well, there are questions which are not answered correctly and got answers that contain sentences unrelated to them. Improvements to the stemmer and specially the morphological analyzer probably result in improvement of performance.
- We observed that documents with more number of user's query terms have higher probability of correct answer matching.

Chapter Six: Conclusion and Future Work

This chapter summarizes our approach to answering list, definition and description questions. It also lists future works for improving the question answering system.

When users need for a certain fact and try requesting search engines for it, they get back a bunch of addresses and snippets which are „related" to their need and it is up to the user to decide which address to choose expecting that the requested fact could be found there. Opening the address could present the user with lots of pages of information and it is the user"s duty to go through the information and extract the actual fact. This is how ordinary search engines help users in need of information. What they do is accept the users" query, search documents in their repository which contains any of the words in the query, rank the retrieved documents and present to the user with title of the page, a snippet, and address (URL) of the page included in the response.

With the help of natural language processing, the information extraction is performed automatically and the user will be presented with answers believed to fulfill the users" request. Question answering systems could use preformatted corpora and provide concise answers in the form of paragraphs, sentences or phrases to natural language questions.

6.1 Conclusion

Question answering system is one of the applications of NLP that provides precise answers to human language questions. QA system for definition and description question could allow someone to know about a term. QA system for list question provides a list of answers for a question. We developed Afaan Oromo list, definitional and descriptional QA system. Afaan Oromo is morphologically rich, so we tried to use a morphological analyzer for simplifying the complexity of words which allows as in generating root words.

We have used a preprocessing technique, in which the data sets were preprocessed using the tasks such as tokenization, case normalization, stop word removal, short word expansion, stemming, lemmatization and indexing allows us to have the same standard between query terms and index terms.

A rule based question classification model were used for classifying users natural language question, which classify users query to their semantic types, queries were generated by removing the interrogative terms after the queries are pre-processed, which allows us to know the kind of information the question is asking for and also to retrieve relevant documents from indexed file. Retrieved documents need to be filtered in order to provide correct answer for the user and we showed how to filter relevant documents from the irrelevant one.

We have used two different methods in extracting answers for list, definition and description questions. The first is a pattern matching (regular expression) method for extracting answers for definition, description and list (where the focus is thing) questions. The other answer extraction method is a gazetter (NER) for answering list question (where the question focus is place).

In order to evaluate the performance of our system we used two criteria. The first criterion was percentage for evaluating question classification component which classified 98% correctly and the other criterion were precision, recall and F-score for evaluating document selection and answer extraction components. The document selection component is tested and scored 0.767 of F-score. The answer extraction component is evaluated with an average precision, recall and F-score, 0.596, 0.723 and 0.653 results are obtained respectively.

6.2 Contribution of the work

The main contributions of this thesis work are outlined as follows:

- AOLDDQAS is the first Afaan Oromo QAS for answering non-factoid question.
- Development of rule based automatic question classification model for AO list, definition and description questions.
- Development of pattern matching rules and gazetters to extract answer of the questions.

6.3 Future Works

Non-factoid question answering is a very complex task, which consumes more time, and needs a number of different NLP tools than factoid questions. In this thesis, we designed

an Afaan Oromo QA that tries to answer list, definition and description question. Below are some of the recommendations we propose for future work:

- Efforts should be made towards improving the Afaan Oromo morphological analyzer.
- Extending this work to other non-factoid questions is an open research area.
- Developing Afaan Oromo WordNet and Word Sense Disambiguation is helpful for the system to better understand user's intension.
- Developing an Afaan Oromo QAS that could perform co-reference resolution would be helpful.
- Integrating Afaan Oromo spelling checker for increasing the performance of QAS can be considered as a future work.
- Improving QAS performance using Set Expansion.
- Developing a machine based question classification method.
- Standard corpus preparation for testing and making experimentation.
- Improving question classification method by using a morphological analyzer on the interrogative terms.

References

[1] Ann Copestake, *Natural Language Processing: part 1 of lecture notes*, 2003, 8 Lectures, (aac@cl.cam.ac.uk), http://www.cl.cam.ac.uk/users/aac/lectures.pdf, Access date: Feb. 12, 2015.

[2] Saad Ahmad, *Tutorial on Natural Language Processing*, University of Northern Iowa, United States, 2007.

[3] L. Hirschman and R. Gaizauskas, *Natural language question answering: the view from here*, *Natural Language Engineering,* vol. 7, no.4, pp. 275-300, 2001.

[4] Alvin Andhika Zulen and Ayu Purwarianti, *Study and Implementation of Monolingual Approach on Indonesian Question Answering for Factoid and Non-Factoid Question, 25th Pacific Asia Conference on Language, Information and Computation*, pp.622–631, 2011.

[5] Vanitha Guda, Suresh Kumar Sanampudi and Suresh Kumar Sanampudi, *Approches for Question Answering Systems*, *International Journal of Engineering Science and Technology (IJEST)*, vol. 3, no. 2, pp.990-995, 2011.

[6] Seid Muhe, TETEYEQ: Amharic Question Answering System for Factoid Questions, Unpublished *MSc Thesis, Department of Computer Science, Addis Ababa University, 2009.*

[7] Desalegn Abebaw Zeleke, LETEYEQ: A Web Based Amharic Question Answering System for Factoid Questions Using Machine Learning Approach, Unpublished *Master's Thesis, Computer Science Department, Addis Ababa University, 2013.*

[8] Hong Sun, Nan Duan, Yajuan Duan, Ming Zhou *Answer Extraction from Passage Graph for Question Answering,2012.*

[9] Burger, J. D. MITRE‟s Qanda at TREC-12, *The Twelvth Text REtrieval Conference, NIST Special Publication SP 500-255,2004*.

[10] Sasha Blair-Goldensohn, Kathleen R. McKeown, and Andrew Hazen Schlaikjer. *A hybrid Approach for Answering Definitional Questions, In Proceedings of the 12th Text REtreival Conference. NIST, Gathersburg, MD 2003 pp. 185–192.*

[11] Masaki Murata, Sachiyo Tsukawaki, Toshiyuki Kanamaru, Qing Ma, and Hitoshi Isahara: *Non-Factoid Japanese Question Answering through Passage Retrieval that is Weighted Based on Types of Answers*. *In the preceedings of the third IJCNLP, Jan 2008.*

[12] Gezehagn Gutema Eggi, Afaan Oromo Text Retrieval System, *MSc Thesis, Information Science Department, Addis Ababa University, 2012.*

[13] Omar Trigui, Lamia Hadrich Belguith, and Paolo Rosso. *DefArabicQA: Arabic Definition Question Answering System, In Workshop on Language Resources and Human Language Technologies for Semitic Languages, 7th LREC, Valletta, Malta, 2010.*

[14] Hang Cui, Min-Yen Kan, Tat-Seng Chua, Jing Xiao, *A Comparative Study on Sentence Retrieval for Definitional Question Answering,* National University of Singapore, 2005.

[15] Debela Tesfaye and Ermias Abebe, *Designing a Rule Based Stemmer for Afaan Oromo Text, International journal of Computational Linguistics*, Addis Ababa, vol. 1, no. 2, pp.1-11, 2010.

[16] Bogdan Sacaleanu and Günter Neumann*, A Cross-Lingual German-English Framework for Open-Domain Question Answering,* Germany, 2006.

[17] Oiry, Morphology, Lecture notes, 2009/Access Date: March 2015

[18] Tilahun Abedissa, *Amharic Question Answering For Definitional, Biographical and Description Questions, Unpublished* Master‟s Thesis, Computer Science Department, Addis Ababa University, Addis Ababa, Ethiopia, November 2013.

[19] Christof Monz *Document Retrieval in the Context of Question Answering*, University of Amsterdam, Netherlands, 2003.

[20] Michael Bendersky, Donald Metzler and W.Bruce Croft, *Effective Query Formulation with Multiple Information Sources*, Washington, USA, February 8–12, 2012.

[21] H. Q. Hu. "*A Study on Question Answering System Using Integrated Retrieval Method"*, Ph.D. Thesis, The University of Tokushima, Tokushima, 2006.

[22] Aberash Tesfaye, *Afaan Oromo Question Answering System for Factoid Questions*, Unpublished *MSc* Thesis, Department of Computer Science, Addis Ababa University, July 2014.

[23] Eric Hatcher, Otis Gospodnetic, michael McCandlles, *Manning Early Access Program,* 2009.

[24] Djoerd Hiemstra, *Information Retrieval Models*, University of Twente, November 2009.

[25] Michael Gasser, *HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya,* Indiana University, July 2012.

[26] D. Moldovan, S. Harabagiu, R. Girju, P.Morarescu, F. Lacatusu, A. Novischi, A. Badulescu, and O. Bolohan. *LCC tools for question answering. In Text REtrieval Conference (TREC)*, 2002.

[27] Mark Andrew Greenwood. "*Open-Domain Question Answering"*, Doctoral Dissertation, Department of Computer Science University of Sheffield, UK, September 2005.

[28] Wesley Hildebrandt, Boris Katz, and Jimmy Lin. *Answering Definition Questions Using Multiple Knowledge Sources*, *In Proceedings of the 12th Text REtrieval Conference* (TREC 2003).

[29] Håkan Sundblad, *Question Classification in Question Answering Systems*, Dissertation of Department of Computer and Information Science, University of Linköpings, Sweden, 2007.

[30] Meiws C.G., "*A grammatical sketch of Written Oromo*", ISBN 3- 89645- 039-5,2001.

[31] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schutze, *Introduction to information retrieval*, 2008.

[32] http://lucene.apache.org/core/, Last Accessed on May 27, 2015.

[33] http://www.cs.indiana.edu/~gasser/Research/software.html, Last Accessed on April, 2015.

[34] Majid Razmara "*Answering List and Other Questions"* Concordia University Montr_eal, Qu_ebec, Canada August 2008.

[35] Patricia Nunes Gonçalves, António Branco "*Open-Domain Web-Based List Question Answering with LX-ListQuestion"* University of Lisbon, june 2014.

[36] http://trec.nist.gov/tracks.html, last accessed on June 20, 2015.

Appendices

Appendix 1: *List of some of Afaan Oromo Short words and their Expansion*

Qar.	Qarshii
Bill.	Billiyoona
Mill.	Milliyoona
A.L.A	Akka Lakkoofsa Awuroopa
A.L.I	Akka Lakkoofsa Itoophiyaa
Ykn	Yookiin / yookan
Kkf(K.K.F)	Kan Kana Fakkaatan
M/B	Mana Baruumsaa
Fkn	Fakeenya
Pirof.	Pirofeesara
Dr.	Dooktoora
I/G	Itti Gaafatamaa
M/Murtii	Mana Murtii
Hosp.	Hoospitaala
M/Ministeeraa	Muumee Ministeeraa
Lakk.	Lakkofsa
Dh.K.D	Dhaloota Kiriistoosin Duura
Dh.K.B	Dhaloota Kiriistoosin Booda
Hogg.	Hooganaa
H/Bulaa	Hoorsisee Bulaa
Q/Bulaa	Qoonaan Bulaa

Appendix 2: *List of some of Afaan Oromo stop words*

Aanee	Gidduu	Itti	Narraa
Akka	Gubbaa	Ittuu	Natti
Akkam	Hanga	Jala	Nu
Akkasumas	Henna	Jara	Nurraa
Akkuma	Hogguu	Sana	Nuti
Ala	Illee	Kan	Siin
Alatti	Immoo	Kana	Silaa
Amma	Inni	kanaafi, kanaaf	Sitti
Ammo	Irra	Kanaafuu	Sun
An	Isaa	Kee	Tanaaf
Ani	Isaaf	Keenna	Tanaafuu
Ati	Isaanirraa	Keessa	Ta"ullee
Bira	Isatti	Keessan	Teenya
Booda	Tun	Keenya	Utuu
Booddee	Iseen	Keessatti	Waan
Dura	Ishii	Kiyya	Warra
Duuba	Ishiif	Koo	Yeroo
Eega	Ishiirraa	Kun	Yommuu
Eegasii	Isii	Malee	Yoo
Fi	Isin	Na	Yookaan
Gama	Isiin	Naaf	Yoom

Kun

Appendix 3: *List of place names*

Amboo	Finfinnee	Galamsoo	Itiyoophiyaa
Dadar	Gursum	Harar	Baabilee
Dassee	Dirre dhawaa	Awaash	Walloo
Ituu	Karrayyuu	Aniyaa	Alaa
Oborraa	Daagaa	Noolee	Jaarsoo
Arsii	Qaalluu	Worjii	Shaifa
Gujii	Boorana	Keeniyaa	Dhedheessaa
Wallaga	Illuu abbabooraa	Samaaloo	Walaabuu
Raayaa	Maccaa	Liiban	Jiddaa
Sirba	Jaawwii	Daal'ee	Dacii
Diigaluu	Gulaalee	Gumbichuu	Gaduulaa
Koonnoo	Yaayyee	Qooqaa	Finca'aa
Gigal gibee	Malkaa waakennaa	Wancii	Xaanaa
Ashangee	Hayiq	Dambal	Abijaataa
Hawaasaa	Langaannoo	Abbayaa	Caamoo
Malkaa	Tulluu	Arsii	Baalee
Hawaas	Maagoo	OOmoo	Afar
Sumaalee	Xuulee	Gudar	Bishooftuu
Adaamaa	Hoolata	Buraayyuu	Huluuqaa
Awaaroo	Ilaamuu	Waddeessa	Walisoo
Meesii	Roobee	Nageelee	Qorkee
Tigraayii	Sumaalee	Afar	Amaaraa
Gaambeelaa	Beenishangul		

Appendix 4: *Sample Test Questions and their Question Ttype*

No	Question	Question Type			
		Definition	Description	List	Unknown
1	Ellaa jechuun maal jechuu dha?	Definition			
2	Vaayirasiin maali?	Definition			
3	Irreecha jechuun maal jechuu dha?	Definition			
4	Siinqeen maalii dha?	Definition			
5	Talaaliin mali?				Unknown
6	Saaphanii sirree maali?	Definition			
7	Katabduun maali?	Definition			
8	Beekumsa jechuun maal jechuu dha?	Definition			
9	Luugni maali dha?	Definition			
10	Aadaan maalinni?	Definition			
11	Algooriziimiin maalinni?	Definition			
12	Pilaaneetooni maal isaanni?	Definition			
13	Urjiin maali?	Definition			
14	Saayinsii umamaa jechuun maal jechuu dha?	Definition			
15	Mooseen maalinni?	Definition			
16	Haroon maali?	Definition			
17	Kooroojoon maali dha?	Definition			
18	Gadaan maali?	Definition			
19	Odaan maali?	Definition			
20	Walaanamuu jechuun maali jechuu dha?	Definition			
21	Hiikni suunaami maali?	Definition			
22	Dhaabanni teeknikaa fi ogummaa maali?	Definition			
23	Hiikni dimookiraasii maali?	Definition			
24	Liiqiin maali?	Definition			

25	Duula jechuun maal jechuu dha?	Definition			
26	Biifooleen maali?	Definition			
27	Irreechi malkaa maali dha?	Definition			
28	Irreechi tullu maali dha?	Definition			
29	Qoqqodama jechuun maal jechuu dha?	Definition			
30	Gadaan roobalee maali?	Definition			
31	Gadaan birmajii maali?	Definition			
32	Ancooteen maalii dha?	Definition			
33	Cacaabsaan maali?	Definition			
34	Marqaan maali?	Definition			
35	Rooketiin maali?	Definition			
36	Footoosinteesiisiin maali?	Definition			
37	Meetaaboliziimiin maali?	Definition			
38	Maayikirooskoopin maali dha?	Definition			
39	Odaan bisil maali dha?	Definition			
40	Odaan nabee maali?	Definition			
1	Algoorizimiin maaliif fayyada?		Description		
2	Faayidaan annani maali?		Description		
3	Faayidaa barumsaa ibsi?		Description		
4	Ga'een afaani maali?		Description		
5	Chaayinaan guudina aduunyaatiif gahee maali qabdi?		Description		
6	Dhadhaan maaliif fayyada?		Description		
7	Gaheen funyaani maali dha?		Description		
8	Ginigilchaan faayidaa akkami kenna?		Description		
9	Gundoon faayidaa maali qaba?		Description		
10	Kilooriniin maalif nu gargaaraa?				Unknown
11	Konkolaataan faayidaa akkami kenna?		Description		
12	Faayidaa maxaaxisaa tareessaa?		Description		
13	Muumeen ministeere gahee maali qaba?		Description		

14	Nyanni madaalamaan maaliif fayyada?		Description		
15	Printeerin maaliif nu fayyada?		Description		
16	Qillensi faayidaa maali keena?		Description		
17	Raadiyooniin faayidaa akkamii qaba?		Description		
18	Faayidaan saayinsii umamaa maali?		Description		
19	Saayinsiin umamaa faayidaan isaa maali?		Description		
20	Siibilii faayidaa akkamittif oola?		Description		
21	Faayidaa teeknoloojii tareessi?		Description		
22	Tembriin faayidaa maali qaba?		Description		
23	Faayidaa tokkumaa tareessaa?		Description		
24	Waraqaan faayidaa maalitiif oola?		Description		
25	Hireen sombaa maali?		Description		
26	Siddaan faayidaa maali qaba?		Description		
27	Faayidaan saatalaayitti maali?		Description		
28	Sangaan maaliif fayyada?		Description		
29	Qadiidaan maaliif fayyada?		Description		
30	Gaheen mar'immanii maali dha?		Description		
31	Faayidaan kannisaa maali?		Description		
32	Ijii maalif fayyada?		Description		
33	Faayidaan aduu maali?		Description		
34	Haydirojiiniin gahee maali qaba?		Description		
35	Gaheen gurraa maali?		Description		
36	Faayidaa guyyaa dubartootaa ibsaa?		Description		
37	Seeliin dhiiga diimaa maalif fayyada?		Description		
38	Baankonni faayidaa akkami kennu?		Description		
39	Faayidaan arkiyooloojii maali?		Description		
40	Faayidaan apiilii maali dha?		Description		
1	Ummanni baarentuu bakka			List	

	meeqatti qoodama?				
2	Oromoon bakka meeqatti qoodama?			List	
3	Qoqqoodama booranaa barreessi?			List	
4	Dammeewwan fiiziksii barreessi?			List	
5	Goosoota seelii dhiigaa barreesa?			List	
6	Diirqama barataa caqasii?			List	
7	Waantoota faalama bishaaniif sababa ta'an tarreessaa?			List	
8	Dhangaaleen nyaataa maal fa'i?			List	
9	Gulli sababa maaliin uumama?			List	
10	Dalagaawwan hiddaa caqasii?			List	
11	Dalagaawwan daraaraa caqasaa?			List	
12	Dammeewwan baayooloojii caqasii?			List	
13	Humni bakka meeqatti qoodama?			List	
14	Maloota faalama qilleensaa ittisuuf fayyadan ibsaa?			List	
15	Karoorri maatii maal irratti hunda'a?			List	
16	Irreechi bakka meeqatti qoodama?			List	
17	Kaayyoowwan fiiziksii ibsaa?			List	
18	Maddoota nyaataa pirootiinii dhaan badhaadhan tarreesaa?				Unknown
19	Seeroota sochii niwutoonii ibsi?			List	
20	Qaamoleen fiizikaalaa maal fa'i?			List	
21	Qabeenya umamaa haaroomfamuu danda'an tarreessi?			List	
22	Sababoota rigata daangeessan barreessi?			List	
23	Sadarkaalee gadaa barreessaa?			List	
24	Fakkeenyoota kaldhabee caqasaa?			List	
25	Afoolli bifa maalittin darbuu danda'a?			List	

26	Akaakuwwan sochii tareessaa?			List	
27	Seeroota yaayyaa oromoo shanan ibsaa?			List	
28	Maddi kaalsiyeemii maal fa'i?			List	
29	Ulaagaalee biiyyaa tarreessaa?			List	
30	Ayyaanni ateetee bakka meeqatti qoodama?			List	
31	Odaan Oromoo bakka meeqqatti qoodama?			List	
32	Faayidaalee siinqee tarreessi?			List	
33	Dalaggaawwan hiiddaa maal fa'aa dha?			List	
34	Akaakuwwan sochii barreessa?			List	
35	Maddi pirootiinii maal fa'i?			List	
36	Naannollee Itoophiyaa warqii oomishuun beekaman tarreessaa?			List	
37	Naannolleen Itoophiyaa horsiisee bulluun beekaman eenyu fa'i?			List	
38	Haariiroowwan hawaasumaa beekamoo ta'an caqasii?			List	
39	Gosa albuudoota qaamni keenya barbaadu tarreessaa?			List	
40	Rakkoolee baay'achuun uumataa fidu?			List	

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of materials used for the thesis have been duly acknowledged.

Declared by:

Name: _____

Signature: _____

Date: _____

Confirmed by advisor:

Name: _____

Signature: _____

Date: _____