.

Addis Ababa University

College of Natural Sciences

Design and Development of Amharic-English Cross Language Question Answering System

Emebet Bekele kebede

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

October, 2019

Addis Ababa University

College of Natural Sciences

Design and Development of Amharic-English Cross Language Question Answering System

Emebet Bekele Kebede

Advisor: ***Fekade Getahun (PhD)***

This is to certify that the thesis prepared by Emebet Bekele, titled*: Design and Development of Amharic-English Cross Language Question Answering System* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies 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: Dr. Fekade Getahun	_____	
Examiner:	_____	
Examiner:	_____	

Abstract

Most of web documents are published in English. Due to this reason, different information retrieval systems that retrieve English documents are developed by different researchers. And, Web users who do not know English are unable to access those systems in their native language. To enable Web users to access web documents written in a different language, a cross-language information retrieval is one of the solutions. So that, a question answering system is one of a practical application of information retrieval. Developing a cross lingual question answering system is essential to enable user to ask a question in a language different than the language in which documents are written. We used a statistical machine translation-based approach for Amharic-English cross language question answering.

In this research work, we have designed architecture of Amharic-English cross language question answering. In doing so, we have developed a statistical machine translation system for translating user question. Also, to make the system more effective, we developed an Amharic named entity tagger with an algorithm of maximum entropy. And by using this system we make a semantic based indexing; it is a great technique to increase precision of the system and pinpointing an answer.

To evaluate the overall system, precision and recall metrics are used. The Experimental result obtained shows that for Amharic-English cross lingual retrieval we got a result of 72% of precision and recall of 79% and for English-Amharic cross lingual retrieval got a result of 65% of precision and recall of 70%.

Key Words: *cross-lingual information retrieval, sematic based indexing and statistical machine translation*

Dedication

To my family, for their support and continuous encouragement throughout my life

ACKNOWLEDGEMENTS

First, I would like to thank God for his blessing and giving me the courage and wisdom to accomplish this thesis. I would like to thank my thesis advisor Dr Fekade Getahuen of the department of computer science at Addis Ababa University for his guidance and encouragement in making this research work. I would like to thank all department of computer science staff members of Addis Ababa University, college of Natural science for their support.

I express my very deep gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

I would like to thank my friend, Mr Ephrem Assefa, for his prayer, advice and encouragement. I am grateful for my classmates and my friends; they were there with me throughout my thesis work.

Finally, I would like to thank gender office of AAU for giving me this scholarship chance to study my MSc.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Acronyms and Abbreviations

API	Application Program Interface
BLEU	Bilingual Evaluation Understudy
CLEF	Cross-Lingual Evaluation Forum
CLIR	Extensible Security Enterprise Search System
EAT	Expected Answer Type
IR	Information Retrieval
MRD	Machine Readable Dictionary
MT	Machine Translation
NER	Named Entity Recognition
NIST	National Institute of Standards and Technology
POS	Part-Of-Speech
QA	Question Answering
RBMT	Rule Based Machine Translation
SMT	Statistical Machine Translation
TREC	Text Retrieval Conference

Chapter One: Introduction

1.1 Background

Question answering (QA) is a system within the fields of information retrieval and natural language processing (NLP), which is concerned with building systems that automatically answer questions posed by humans in a natural language [1] . It is a particular type of search engine that allows users to ask question in natural language and will retrieve precise and accurate answers.

The rapid growth of the Internet has made a vast amount of textual information available online. In information retrieval when users insert query, links of different documents that have huge amount of relevant and non-relevant documents will be retrieved, so that users have to skim the retrieved document to get the desired information. It is difficult to skim this huge information. One solution to mining information of interest from these vast electronic resources is provided by textual QA. It is a technology that returns short and exact answers to questions expressed in natural language. The answers are extracted from electronic documents on web or from local intranet collection.

In contrast to a standard information retrieval system, in QA system queries are usually well-formed natural language query clauses (instead of a set of keywords), and the identified answers should be textual fragments representing the answer (instead of complete documents containing the answer). The QA system is expected to present the correct answer. To do so, it requires NLP techniques to understand what the user asks and to select certain string as a correct answer from the retrieved document. A question answering system has an additional component to analyze the user query and documents using NLP tools or other techniques.

START, the world's first Web-based QA system, has been publicly accessible and continuously operating since December, 1993 [2]. It constructs its answers by querying a structured database of knowledge or information, called a knowledge base. A user can query the system in English. The query’s analyzed form is matched against the knowledge base to retrieve the stored knowledge. The system will then produce a response in English.

The increasing of local documents on the web led researchers to work on local QA systems. Local QA system allows users to ask questions in their own language, querying of local electronic documents and to get the answer on their own language. Existing Amharic QA systems allow users to ask questions in Amharic and get answer in Amharic [3, 4]. In such cases, the system is querying only Amharic electronic documents.

However, the web contains documents in different languages, local QA system in the case of under resourced languages such as Amharic is not sufficient to get desirable answer. To address this problem, multi or cross language QA system is one solution. A cross language QA system allows the users for querying in a language different than the language in which the documents are written.

TREC (Text REtrieval conference) and CLEF (Cross-Language Evaluation Forum) have a great contribution in improving the performance of cross language information retrieval research areas. TREC is an ongoing series of workshops focusing on a list of different information retrieval (IR) research areas, or tracks. Its purpose is to support and encourage research within the information retrieval community by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies and to increase the speed of lab-to-product transfer of technology. Whereas CLEF is a self-organized body whose main mission is to promote research, innovation, and development of information access systems with an emphasis on multilingual and multimodal information with various levels of structure [5].

Even though there are a number of researches conducted on bilingual Amharic- English search engine [6], to the best of our knowledge there is no work related to bilingual Amharic- English QA system. The purpose of this thesis is given a natural language query in Amharic or English language, to find answers in textual documents written in another language (Amharic or English). The proposed system increases the probability of getting relevant answer by searching question from Amharic and English electronic documents, so that most web documents are published in English.

1.2 Motivation

English is a language which has a long age on the web, most web documents on the web are published by English language. Web users who do not know English language should not be neglected. Amharic QA systems are limited to querying Amharic documents. Those systems are not sufficient to get desirable information.

1.3 Statement of the Problem

The diversity of documents on the web leads researchers to work on multilingual information retrieval. Isozaki *et al* [7] propose English-Japanese cross language QA system. They used dictionary-based word translation and web-based back transliteration approach for query translation. NLP Techniques are used as well for indexing and answer extraction. Neumann and Sacaleanu [8] also propose Cross–Language Question/Answering–System for German and English. However, their approach receives only German language query, parses and translates it into English, and searches for answers in a large English text collection maintained by the full–text search engine.

The existing Amharic QA systems [3, 4] allows users to ask question in Amharic language and it will retrieve precise and accurate answer in Amharic as well. The answer is extracted from only documents that are written in local language and because of that it has a limitation of returning answer from documents in English language.

Besides, the existing Amharic QA systems use shallow techniques instead of NLP tools for question and answer analysis. NLP techniques are required in QA to understand what the user needs and to extract answer from retrieved documents. NLP tools like POS tag and Named Entity Recognition have a great role to identify the exact answer from documents. Due to this reason those system are less effective in comparison to other QA system [4].

1.4 Objectives

General Objective

The general objective of this study is to design and develop a generic Amharic-English Cross language QA system.

Specific Objectives

To realize the aforementioned general objective, the following specific objectives are identified

- Review related works.
- Develop an Amharic named-entity tagger system.
- Prepare training data set and develop a classifier model to classify a question type.
- Develop a statistical machine translation model for translation purpose.
- Design a model for cross language QA system.
- Develop a prototype for cross language QA system
- Evaluate the proposed system.

1.5 Methods

In order to achieve the objectives of the research, the following methods will be used

Literature Review

In order to get deep knowledge about the topic area, reviewing of related literatures is vital. We will conduct literature review on techniques and algorithm used for developing different local and foreign QA systems. In a cross-language information retrieval system, translation is the most basic component to translate query language to document language. There are different alternative approaches for language translation. We will conduct a review on alternative approaches of translation.

Data Collection

Data that will be used for testing prototype will be collected from different sources. Parallel corpus is the most essential data for modeling a statistical machine translation model. To make the translation model effective, increasing the corpus and collecting from different domain is vital. To achieve this, a large amount of Amharic and English corpus (56,044 parallel sentences) are collected from different sources. i.e. Bible, constitution, news and history documents. We will also collect a corpus for developing an Amharic named-entity tagger. First, we will collect a document, and then we annotate the documents for training a named-entity model.

Design and Development of Prototype

In design phase, the overall architecture of the proposed system will be designed. In order to evaluate the proposed method, a prototype will be developed using programming tools and open source. We will use JAVA programing language to develop a proposed system.

The following are open source libraries and platforms that will be used to develop the system.

MOSES decoder [9] is an open source toolkit for building and running a statistical machine translation system, which is utilized for this study. It is most widely used for implementing SMT approaches. MOSES toolkit is selected as it is open source, easily customizable and provides full control over translation process. Moses is a system that allows training automatic translation models for any language pair.

IRSTLM [10] is a language model toolkit, features algorithms and data structures suitable to estimate, store, and access very large N-gram language models. It is easily integrated into a popular open source called Moses.

GIZA++ [9] is a toolkit to train word alignment models. Word alignment is mapping of words between two sentences that have the same meaning in two different languages.

Apace Lucene [11] is a high-performance, popular, open source, full-featured text index, and search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search.

Apache Solr [12] is an enterprise search engine that's optimized to search large volumes of text-centric data and return results sorted by relevance. It is scalable, ready to deploy and build on Apache Lucene,

Opennlp [13] is an open source Java Library, it is a machine learning based toolkit for the processing of natural language text. OpenNLP API has a class to implement different NLP tasks like sentence detection, tokenization, named entity recognition, part of speech tagging etc. opennlp tools have a model for some languages including English to do NLP tasks. We employ English NLP models for NLP tasks that are necessary for our system. For languages that are not supported by this tool, we can train a model using our data set using Opennlp trainer class. So that we can also train and evaluate our own models for any of the NLP tasks using Opennlp trainer. We will use Opennlp trainer class for generating a model for Amharic NLP tasks.

Apache Tika [14] is a toolkit that detects and extracts meta data and text from over a many different file types (such as PPT, XLS, and PDF). A language detector is one of the components in Apache Tika toolkit to detect language of documents. This component customized to detect Amharic document and will be used in our system.

Evaluation

The proposed work will be evaluated using different metrics. The metrics are precision and recall metrics to measure the performance and efficiency of the system. For SMT models, an automatic metrics called Blue score will be used.

1.6 Scope and Limitations

Scope

The scope of this research work is limited to extracting answers from documents that are written in English and Amharic language. This research work is limited to factoid questions.

Limitations

The proposed system also takes only text electronic documents as a data source. Different multimedia documents like audio and video are not supported in this thesis.

1.7 Application of Results

A question answer system has a great contribution in different real-world applications such as automated customer services, suggests directions in driving system, E-learning, collaborative learning, reservation system and help desk system. This research work enables to implement those real-world applications appropriate to Amharic language user even if the source document is written in English.

1.8 Organization of the Thesis

The remaining part of this thesis is organized as follows. Chapter Two presents the review of literature in the domain. Chapter Three presents related works done on QA and cross language information retrieval system and related areas. The fourth Chapter deals with the design of Amharic-English cross language QA system. Chapter Five presents the implementation of the Amharic-English cross language QA system and experimental results. Finally, conclusions and recommendations are given in Chapter Six.

Chapter Two: Literature Review

2.1 Information Retrieval

The need to store and retrieve written information became increasingly important over centuries, especially with inventions like paper and the printing press. Soon after computers were invented, people realized that they could be used for storing and automatically retrieving large amounts of information. Search and communication are the most popular uses of the computer. Recently, for most people's searching for information on the Web is a daily activity [15].

Information retrieval [16] is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information. In IR a user can find the most relevant documents which partially match a certain request, from a large set of documents (e.g., a large database of documents or the Web). Therefore, IR addresses the problems associated with the retrieval of a subset of documents ordered by decreasing likelihood of being relevant to the given query from a collection of documents.

A Question Answering system [1] aims at returning a concise answer rather than a list of relevant documents to the user's question. It takes natural language queries instead of keywords and returns the exact answer instead of list of documents that contains answer. Question answering systems are classified into two groups [1]: closed domain and open domain. Closed domain QA systems are restricted to a specific domain (like medical, music automotive maintenance etc.). These systems exploit a domain specific ontology. Whereas, an open domain QA system deals with questions about nearly everything and can use universal ontology and information such as World Wide Web.

Depending on the focus of question a QA system can be also classified as [43]:

- **Factoid questions:** questions that can be answered with simple facts expressed in short text answers. such as who, what, when and where questions
- **Temporal question answering:** the automated interpretation of questions with temporal element such as absolute or relative times, points, durations and extraction of answers with temporal aspects

- **Spatial question answering:** providing answers to questions involving spatial objects (e.g location, regions), attributes (e.g size, shape) and relations (e.g above/below, inside/outside. Near/far), possibly requiring spatial interface.

- **Definitional question answering:** the automated creation of definition and description of objects or terms.

- **Biographical question answering:** the automated creation of answers regarding questions about the most significant characteristics and events in the life span of a person, group, or organization.

- **Opinioned question answering:** the automatic detection of opinions (of individuals, group, or institution) and response to question about viewpoints and perspectives often times regarding subjective content.

- **Multimedia/ Multi-modal question answering:** processing queries that might be expressed a range of media (text, audio, imagery, video) and/or modalities (i.e auditory, visual sense) and extracting responses from sources that may equally be in a range of media and/or modalities.

- **Multilingual question answering:** answering questions either for users with varying languages or from multilingual sources requiring trans lingual retrieval, foreign language content extraction, and language specific response generation.

2.2 General Architecture of Question Answering System

A typical QA system has four main components. Those are question analyzer, document retriever, passage retriever and answer extraction. Figure 2.1 [17] shows a general architecture of a QA system.

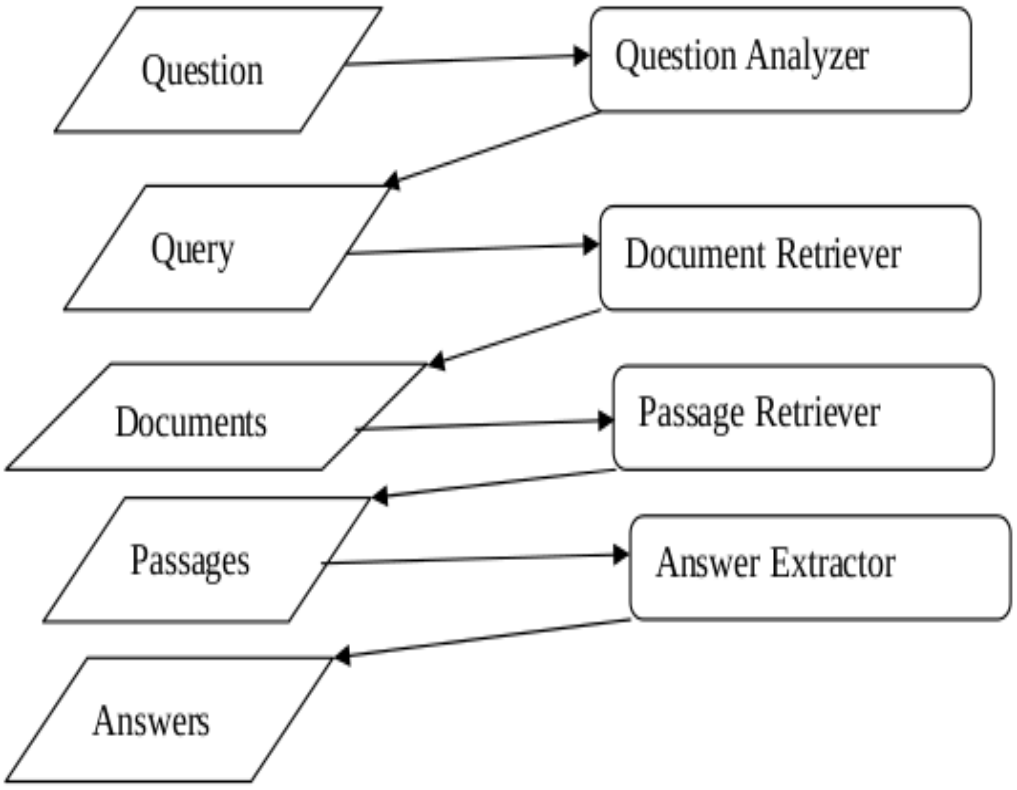


Figure 2.1: General architecture of question answering system

2.2.1 Question Analyzer Module

A question analyzer is the first component in which the system starts to analyze the natural language question posed by the user right from the input window in a web browser. It is an analysis of question structure and it's semantic. Before a question can be answered by QA system, it must be understood at several different levels. The questions or user needs to be identified based on a combination of syntactic, semantic and pragmatic knowledge. It is the most important component of QA systems. The following are key tasks to be performed in the question analysis stage:

1) Question Classification

Questions can be classified as questions of type place, person (Entity), term definition, quantity, listing, explanation, True/False, Time, Choice, and so on. Based on the question type, an expected answer type will be identified. The three main approaches to classify a question can be: rule-based, machine learning based and language modeling based.

- **Rule based approach:** hand-written grammar rules and a set of regular expressions are employed to parse a question and to determine the answer type [3]. This approach determines a question type based on the sentence pattern, which includes the interrogative word, certain sequences of words and some representative terms of particular question classes. Rules can be constructed manually to automatically classify questions. However, this approach has limitations; it is difficult to include all the possible patterns of the language in the rules and difficult to construct rules [4].

- **Machine learning approach:** the type of the question is predicted after the machine is trained with a training data set. However, it needs more training data in order to provide best results because it automatically learn and improve with experience [4].

- **Language model approach:** a language model assumes human language generation as a random process and its aim is to represent that process by a statistical model to predict the next word using context of the previous word. In the question classification task, a language model is prepared for each question type of sample questions. When a new question Q comes, the probability P(question|questiontype) for each question type is

calculated and the one with the highest probability is picked as the type for the new question Q [18].

2) Question Focus Determination

The question focus is a phrase or word in the question that can help to disambiguate it and together with the question stem (e.g., who, how much, where) can help to deduce the expected answer type (EAT). It identifies a specific aspect of the question topic. The question topic defines the semantic domain or the context in which information is wanted.

For example, Q: In which month is Meskel celebrated?

Question Focus: which month

3) Expected Answer Types Determination

The expected answer types seem similar to the question types but these are the types or semantic category expected from the candidate answers for the given question. Determination of semantic category of the expected answer will help for both candidate passages retrieval and the actual answer extraction. The recognition of the expected answer type is done by:

1. Identifying the question word/phrase that indicate the semantic category of the expected answer (e.g., when, where) and

2. Mapping this word or phrase against an off-line ontology of possible answer categories. This means mapping “WHO” or “ማን” question word to “PERSON” answer categories. and mapping “WHERE” or “የት” question word to “LOCATION” answer categories.

In the case of factual questions, most of the time the expected answer type is identical to the question focus. For complex questions, the focus and the expected answer type are different [20].

4) Query Construction

Another task of the question analysis component in QA system is formulating an appropriate query from the user's natural language questions and sends to the document retrieval. Query construction can be done by tokenizing, stemming, normalizing, stop word removing and query

expansion. A query expansion based on words in the question is done by adding synonyms of words in question as a query term. This help the system to retrieve related documents. And also query expansion based on expected answer type is adding the expected answer as a query term. This technique needs creating a semantically based index of documents based on expected answer type. This means, documents will be analyzed semantically by labeling words in the document as location name, organization name, person name, numerical, date, etc. Named entity tagger is used for labeling words by identification of named-entity of words. This technique is effective as it is highly essential to extract answer from the retrieved document [19].

2.2.2 Document Retriever Module

Document retrieval is the task of identifying potential documents which have a potential to contain an answer for the given question. The query derived from the question analysis component will be an input to this component. In order to achieve the objective of a QA system in an efficient manner, it is a common and logical practice to use an IR system as a front-end. This helps to concentrate on ranked list of documents rather than a large collection of documents to easily manage [20]. The retrieved documents which are believed to contain the answer will be passed to the next component, the passage retrieval component.

Some QA systems use the IR system to retrieve relevant documents where further processing remains to be the task of subsequent components such as passage retrieval/answer extraction system. And some of QA systems use the IR subsystem to directly perform the passage retrieval/answer extraction itself.

2.2.3 Passage Retriever Module

The passage retrieval selects relevant passages of text from the candidate documents retrieved. Passage retrieval aims to find the text passages that may contain the exact answer of the given question. It has long been studied in IR and recently has been an important component in QA systems. Although QA systems aim at providing exact answers only, researchers show that users prefer paragraph level chunks of text with appropriate answer highlighting [21]. This enables users to pick the appropriate answer when a question seems to be ambiguous.

2.2.4 Answer Extractor Module

Answer extraction also known as answer selection/pinpointing takes query and answer type determined from question analysis, then finds an answer from the retrieved documents or passages. As search engines return a list of ranked documents, answer extraction will return a list of ranked answers. There are four approaches for answer extraction [22].

- **Heuristic** This approach approximates matching between the question and the answer. It is done by counting the number of term matches and computing measures such as average distance of query term to answer term, density of all matching terms and so on, and combining these factors using weights learned in training.
- **Pattern-based** In this approach, extracting correct answers will be done with the help of matching patterns. The patterns for the answer will be formulated after analyzing a number of questions with their answers. These approaches emphasize shallow techniques over deep NLP.
- **Relationships** This approach employs linguistic knowledge to compute candidate answer scores. Syntactic and semantic processing are performed both on the question and each candidate passage or answer. Candidate answers are scored based on finding the same relationship.
- **Logic-based:** This type of approach uses theorem-proving, although not necessarily in a rigorous manner. The basic technique is to convert the question to a goal, then for each passage found through conventional search methods, convert it to a set of logical forms, representing individual assertions.

2.3 Paradigms to Question Answering System

There are two major modern paradigms to QA, focusing on their application to factoid questions [23]:

2.3.1 IR-Based Question Answering

IR-based QA or sometimes text-based QA relies on the enormous amounts of information available as text on the Web or in specialized collections. This paradigm employs information retrieval technique to extract passages directly from these documents, guided by the text of the question. The method processes the question to determine the likely answer type (often a named entity like a person, location, or time), and formulates queries to send to a search engine. The search engine returns ranked documents which are broken up into suitable passages and reranked. Finally, candidate answer strings are extracted from the passages and ranked.

2.3.2 Knowledge-based Question Answering

In this paradigm, an enormous amount of text on the web must be encoded in more structured forms. The database in which the answer is extracted can be a full relational database, or simpler structured databases like sets of RDF triples. A semantic representation of the query is constructed. Then the answer for a given natural language question is retrieved by mapping it to a query over a structured database. The meaning of a query can be a full predicate calculus statement.

BASEBALL [24] is a knowledge based QA that answers questions from a structured database of baseball games. This system uses a semantic parser for mapping from a text string to any logical form. Semantic parsers for QA usually map either to some version of predicate calculus or a query language like SQL or SPARQL, Table 2.1 shows how a question is represented in a logical form.

Table 2. 1: Logical form of a question

Question	**Logical form**
When was Ada Lovelace born?	birth-year (Ada Lovelace, x)

2.4 Search Engine

A search engine is a tool that allows a user to enter keywords and retrieve information on websites contained in its catalog or database. It is a practical application of information retrieval techniques. search engines can be found in different application, such as web search engine, desktop search engine or enterprise search engine and QA [16].

Major Components of Web Search Engine

Generally, search engines contain three components:

- Crawler component
- Indexer component
- Query engine component

A Crawler Component has the primary responsibility for identifying and acquiring documents for the search engine. There are a number of different types of crawlers, but the most common is the general web crawler. A Web crawler is an automated program which automatically traverses the Web by downloading documents and allowing links from page to page. A web crawler is designed to follow the links on web pages to discover and download new pages.

An Indexer Component collects, parses, and stores data to facilitate fast and accurate information retrieval. The purpose of storing an index is to optimize speed and performance in finding relevant documents for a search query. Without an index, the search engine would scan every document in the corpus, which would require considerable time and computing power. The indexer component extracts all the words from each page (parsing), and records the URL where each word occurred. The result is a large lookup table that can provide all the URLs that point to pages where a given word occurs.

A Query Engine Component accepts user queries (keywords) for a particular topic and looks up the index component. The document in the index that matches with the keyword will be selected and displayed to the user. The query engine component has the task of sorting the results in a way that results near the top are the most likely ones to be what the user is looking for.

2.5 Cross Language Question Answering System

Searching for information is our daily activity in this information era. Preferably, we are interested in finding information written in our native language. However, relevant information is not always available in our native language, because of that we are forced in finding information written in other foreign languages in many situations. This gives rise to the development of cross-language information retrieval and cross language QA system, whose goal is to find relevant information written in a different language [25]. A cross language QA system allows the user for querying in a language different than the language in which documents are written.

2.5.1 Query Translation vs. Document Translation

In a query translation approach, users query is translated to document language. Where as in document translation approach, documents are translated to user query language. It is generally believed that query translation is the most appropriate approach [25], since query translation approach in a given a query, the user is allowed to choose the languages of interest, and the query can be translated into the desired languages. In case where the user is capable of understanding the translation(s) of the query, he/she will be able to correct the translation before it is used to retrieve documents. This approach is flexible and allows for more interactions with the user [25].

However, query translation often faces the problem of translation ambiguity and this problem are comes due to the limited amount of context in short queries. From this perspective, document translation seems to be capable of producing precise translation due to richer contexts [25]. The availability of efficient MT systems able the document translation approach possible.

Several studies have compared the query translation and document translation approaches using the same translation systems. For example, Oard and Hackett [26] propose Document Translation approach for Cross Language Text Retrieval (English-German). The authors aim is to evaluate which approach is more effective to get appropriate document. Three classes of queries are prepared to evaluate the system: short, title and long queries depending on user query. This research shows us query translation and document translation perform about equally well on title and short queries. But, in long queries document translation is more effective in comparison to query translation, so that a lot of words in the document help the system to get linguistic context

and to select the most appropriate document, but document translation is tedious and takes a lot of time to translate documents. Due to the limited advantage of document translation shown in experiments, most current research and development on CLIR use query translation due to its high flexibility.

2.5.2 Approaches to Translation in CLIR

In information retrieval system, word ambiguity or multiple representations of a meaning makes difficult to match the relevant document against query. This matching process will become more difficult when language of the document and query are different. In addition to the monolingual IR problems, translation is the main problem for CLIR and MLIR. There are three classes of approaches to translation: MT system based, dictionary-based and parallel corpora based approaches [25].

2.5.2.1 Machine Translation Approach

Machine translation systems are constructed for the primary purpose of providing full-text translation without any manual intervention. The basic approach to use a machine translation system in CLIR is simple: one just has to submit either the query (in query translation strategy) or the documents (in document translation strategy) to an MT system to obtain a translated version. Then the translated version is used as a query in monolingual IR. Generally, MT systems are classified into two categories: traditional rule-based MT and statistical MT (SMT).

1. Rule-based systems operate using rules and resources constructed manually. Rules and resources can be of different types: lexical, phrasal, syntactic, semantic, and so on. Rule based MT approach is expensive to create, maintain and adapt since the rule-based MT uses an integrated linguistic knowledge, rules and resources of both the source and target languages. The linguistic knowledge includes tagging, parsing, morphology, syntactic, semantic, and lexical knowledge of the source and target languages. The linguistic rules comprise rules for analyzing, transferring (including syntactic, semantic & lexical), and generating the source and/or target languages. Particularly, it is a challenge to develop MT using rule based approach for NLP scares resource language like Amharic [27].

2. SMT is built on statistical language and translation models, which are extracted automatically from a large set of texts and their translations (parallel texts). The statistical approach relies

heavily on bilingual parallel aligned corpora of the source and target languages. From such parallel texts, various types of translation relationship can be extracted and used to translate new texts. The challenge is minimized since the statistical based approach requires very limited computational linguistic resources compared to the rule-based approach that might take so many years to develop some or all of the mentioned linguistic resources.

2.5.2.2 Dictionary-Based Translation

This approach employs machine-readable dictionaries to identify and select the possible translations of each source word. The quality of translation is strongly dependent on the quality of the dictionary, including the correctness and the completeness (or coverage) of the translations included. Coverage can be improved using stemming techniques, although they tend to increase the level of uncertainty since more words with different meanings are conflated into the same stem particularly it is a challenge to use this approach for Amharic language which is considered as one of morphologically complex language [27]. In addition, dictionary-based approach translates single word at a time. Some phrases and compound words might be wrongly translated while they are translated separately.

2.5.2.3 Parallel Corpora Based Translation

This approach exploits a parallel corpus for translating source to target language. A parallel corpus contains both source language texts and their translations in the target language. Using these corpuses, we try to extract the strong translation relations between the two languages, either at the word level or at a higher level (e.g., phrase level). These translation relations can then be used to translate queries or documents.

Chapter Three: Related Work

In this chapter, review of related research works in the area of QA and cross language QA system will be explained. We will discuss monolingual QA systems from point of view of approach used for question classification, answer extraction and tools used for document retrieval. We will also discuss cross language QA system from point of view of translation approach used.

3.1 English Question Answering System

Katz *et al.* [2] developed the first web based QA system called **START.** It has been publicly accessible and continuously operating since December, 1993. It uses knowledge-based approach. The knowledge base is constructed from a set of information resources that is hosted locally or accessed remotely through the Internet and it contains structured, semi-structured and unstructured information**.** START employs a natural language annotation to match question to candidate answers. Natural language annotations are computer-analyzable collections of natural language sentences and phrases that describe the contents of various information segments. The annotation enables the system a very high precision and also makes possible indexing of non-text resources which cannot be analyzed by other system.

However, START QA system is language dependent and only appropriate to the English language. In this system, a user can query the system only in English language and the query is analyzed in the same way as assertion to create knowledge base. The system matches query's analyzed form with knowledge base and will retrieve response in the English language.

A simple factoid QA system was implemented by Sreelakshmi and Jama [28] using the technique called Semantic Role Labeling. This technique is used to label the semantic roles in a sentence and employ to answer extraction component in order to pinpoint the answer from the retrieved document. The semantic role labeler system identifies all the constituents that fill a semantic role, i.e. to determine the roles like agent, patient, instrument, location, time etc. in a sentence. Semantic role labeling (SRL) has been used in several stages of automated QA systems. The authors used SRL for identifying the exact answer of a question in the answer extraction stage to improve the performance of QA. For example, a question that starts with "Who" uses agents or

actors. i.e. persons as an answer. Similarly, ‘Where’ questions select a location as an answer and ‘When’ questions select time or temporal tags as answer.

In indexing stage also SRL can be used for labeling words in a document. Augusto and Molla [29] used SRL in indexing stage. In QA systems, indexed document that contains information about semantic relation among words enables the system to be efficient. In addition to co-occurrence proximity to each other, information about semantic relation enables to extract answer from documents easily and efficiently. A semantic relation index allows the retrieval of the same piece of information when queried using syntactic variations of the same query such as: “Bill kicked the ball” and “The ball was kicked by Bill” have different structure but have the same meaning.

Even if SRL is used to improve QA system, a semantic analysis of this sort is at a lower-level of abstraction than another NLP tasks such as named entity, syntax tree. For a question answer type of “organization” will be difficult to detect organization name from documents. For such a case it is difficult to pinpoint the exact answer.

Mihalcea and Moldovan [19] proposed an approach, which is a semantic based indexing approach for retrieving relevant document. The author precedes NE tagging from document indexing. In query processing stage answer type taken as a query for retrieving documents. Documents which do not contain both the keywords and the answer type will be discarded. This approach increases the precision of QA system by semantic based indexing. Their experiment shows that the proposed approach will retrieve a document which contains an answer in a better way in comparing with traditional IR. From this work we learn that, NE tagging prior to document indexing has a great role in increasing precision of the system.

3.2 Amharic Question Answering System

Seid Muhie and Mulugeta Libsie [3] developed the first Amharic QA system called ASK (TETEYEQ). The author uses handcrafted rules to identify question and answer types. The document retrieval component of the system uses Lucene contribution packages called SpanNearQuery and RegexQuery. Those packages help to consider the distance between the

query terms and the expected answers to select a relevant document. This enables the system to maximize retrieval of documents with possible answer particles present.

The experimental result shows that the question processing module correctly classifies 89% of the questions using the rule-based classification.

Desalegne Abebaw [4] developed a Web based Amharic QA system for factoid questions. The major difference of this system from Seid Muhie and Mulugeta Libsie research work [3] is the question classification approach and the data source used. The author employs an automatic question identification by using machine learning approach to improve the performance of the system. The system classifies questions or to predict expected answer into one of the four question types (“person‟, “time‟, “place‟, and “quantity‟). Support Vector Machine (SVM) approach was used for classifying question type.

The experimental result shows that question processing module correctly classifies 94.2% of the questions by employing Support Vector Machine (SVM) approach.

3.3 Cross Language Information Retrieval System

Amharic English bilingual search engine was developed by Mequannint Munye Zeru and Solomon Atnafu [6]. It allows user to formulate query in one language (Amharic or English) and get the Web search results in both languages. The authors employ dictionary-based approach for translating query language to document language. To implement this approach, bilingual Amharic-English dictionary and transliteration are used. Query and document preprocessing are language dependent tasks; those processes are done separately depending on the language. Because of different characteristics of the two languages (Amharic and English), existing Amharic and English search engines are used for the system. They used Amharic web search engine that was developed by Tessema Mendaye *et al* [30] and English search engine developed by Arasu *et al* [31] was used. In this system, detecting proper noun is done by checking whether the word is found in dictionary or not. But some proper nouns are found in dictionary and they might be translated wrongly. Proper nouns must be translated by transliteration. So that the algorithm used to detect proper noun is not effective. Another limitation also the Amharic search engine that was adopted for this work was done by language specific crawler. The crawler use

meta data information to detect the language of web documents. However, this language identifier method was not effective and cannot detect web documents that does not include meta data information.

To test the effectiveness of Amharic and English query translation component, manual translation of Amharic and English queries is used. Their experiment shows that in English query translation 93.33% of the queries are properly translated and Amharic query translation 88.57% of the queries are properly translated. Amharic English bilingual search engine has a limitation in using dictionary-based approach. That is the quality of translation is strongly dependent on the quality of the dictionary, including the correctness and the completeness (or coverage) of the translations included [25].

Isozaki *et al* [7] propose Japanese- English cross language QA system (SAIQA-J/E). The authors adopted question analysis module of Japanese QA system developed by Isozaki *et al* [7] and answer extraction/evaluation modules of English QA system developed by Ikehara *et al* [32]. The author develops English NE recognition using hand-crafted rule-based method since NE recognition has great contribution in answer extraction. The major tasks introduced in this work is Japanese-English query translation module. This module has a responsibility of translating the output of the Japanese question analysis module into English. Japanese-English dictionary is used to translate Japanese query term. If query term is not found from the dictionary, transliteration is used for translating query term.

Neumann and Sacaleanu [8] developed a German-English question answering system known as BIQUE. It is a system that receives a German language query, parses and translates it into English. Then it searches for answers in a large English text collection maintained by the full–text search engine. The authors use a public-domain full-text retrieval engine that is called MG system. It is selected as it is an easy usable software package that can handle text corpora of several Gigabytes very efficiently. Queries and documents are linguistically analyzed using different integrated NLP tools like tokenizer, analyzer of compound words, part–of–speech tagger, morphological analyzer, named entity recognizer and chunker for phrase recognition. Also queries and documents are uniformly represented as weighted sets of structured (possibly linked) objects in order to facilitate a robust and efficient comparison between queries and answer candidates.

For translating German language question to the English language, machine translation approach is utilized. In this system to fill the gap of coverage three different translation services have been used: FreeTranslation, Altavista and Logos. The result released by the QA track show that the system retrieves 14.5% correct answers out of test set questions.

Haddade *et al* [33] developed a bilingual French-English QA system called CINDI_QA. This system receives a French question and returns the answer to that question in English. The authors developed a template based bilingual QA system. This template is defined as a model or pattern using for making multiple copies of a single object. It is a high-level construct in the field of computer science that represents several similar entities by identifying general structure that matches those entities. The author defined six templates to identify best answer and the template module checks the input questions to see if it matches one of the six templates already defined. Components of this system are independent and it is possible to be replaced by other similar tool without altering the design of the system. Google translate was employed for translating French question to English. Even if Google translate is a system developed with SMT based approach that translates different languages including Amharic language effectively, it is not an open source and impossible to integrate to our system.

3.4 Summary

Question answering systems have been developed by different researchers in different approaches. We reviewed monolingual, cross lingual and bilingual QA system that are done in different languages. we discussed the monolingual QA system from the perspective of approaches that are used in their component. The cross lingual QA systems are discussed from the perspective of translation approaches they used such as dictionary and machine translation-based approaches. All cross-language QA system that use dictionary-based approach have a limitation of the correctness and the completeness (or coverage) of translations. Since they are strongly dependent on the quality of the dictionary. Also, to translate compound word or phrases, the system translates separately word by word which might be translated wrongly. This limitation can be solved using machine translation approach.

Cross language QA systems that are developed for different languages are available. But, due to the difference in morphology, syntax and structure of a sentence of Amharic language with other languages, they do not effectively work for Amharic language.

Chapter Four: Design of Amharic-English Cross Language Question Answering System

4.1 Overview

This chapter presents the architectural design of Amharic-English cross language QA system, the main components and their interaction. The major components of an Amharic-English cross language QA system shown in Figure 4.1 are *web crawler, indexer*, *question translation*, *question analysis, passage retrieval, answer extraction*.

The web crawler discovers and downloads documents from the web. In the indexer component, document language identification and analysis of documents depending on their language are performed. The language identification module identifies documents language whether Amharic or English. This module is essential in indexing phase to make document analysis tasks separately, since it is language dependent. Named entity tagging is an important task in indexing stage particularly, in factoid QA system whose answers are named entity. It enables to extract answer from the relevant documents. After document analysis processes are done, the documents will be indexed for fast searching.

The question translation component translates the user query from source language to the destination language to retrieve answers from both Amharic and English electronics documents. Both the translated and the original questions will be inputs to the question analysis component and each are analyzed separately. The question analysis component has sub components: question parsing, expected answer type determination and query generation.

The question analysis component analyzes a user question and derives a query appropriate to passage retrieval component. The query derived from question analysis component will be input to passage retrieval component. The passage retrieval component retrieves a set of relevant passages that contain answers by matching keywords and expected answer with document index. Finally, the answer extraction component retrieves precise answer from the relevant passages by constructing windows around query term, calculates their weight and ranking depending on their weight.

The remaining part describes each of the sub-components in detail.

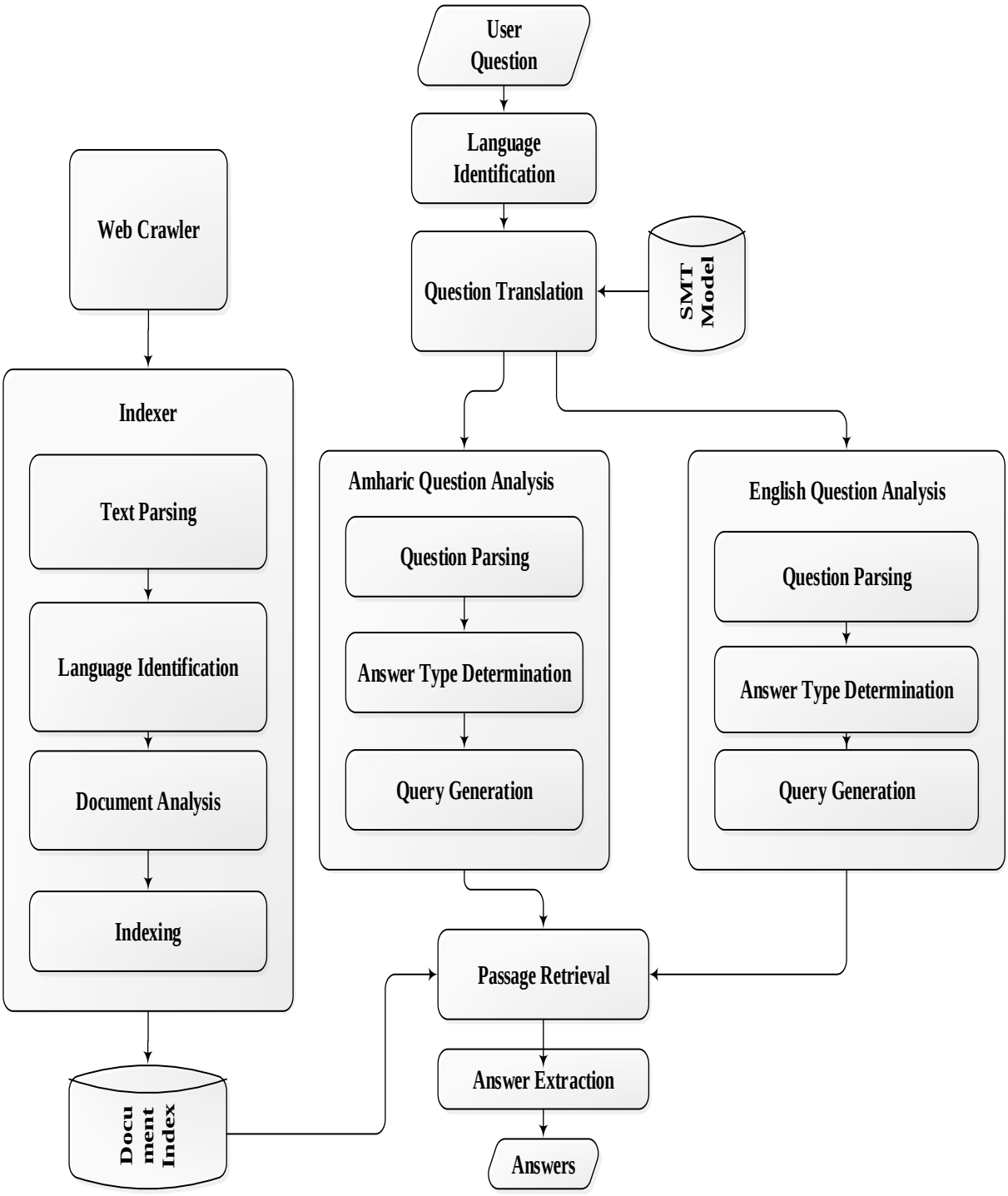


Figure 4.1: General Architecture of the Proposed System

4.2 Components of the Architecture

4.2.1 Web Crawler

The crawler component has the responsibility for identifying and acquiring documents for the search engine. A web crawler is designed to follow the links on web pages to discover and download new pages. The crawler finds document by beginning at one page called seed and following all of the links on that page. In this thesis, web documents used as a data source for retrieving answers to user's question. Amharic and English web pages are given as a seed. This enables the system to acquire both Amharic and English documents, since both documents are processed in our system.

A crawler possesses certain features to build a web text retrieval engine. The crawler fetches different document formats like HTML, XML, doc, pdf. These documents are fetched from the Internet and parsed individually using text parser to get the clean content. We selected sites for the seed by considering the language of the websites. We selected a site that contain a data in Amharic and English language for the seed. Those are

- https://www.bbc.com
- https://www.zehabesha.com
- https://mereja.com
- https://www.ethiopianreporter.com

4.2.2 Indexer Component

The indexer component creates a representation of documents that is appropriate for fast searching. In this component, the main tasks to be performed are text parsing, identifying documents language, analysis of documents depending on the language of documents and creating index.

4.2.2.1 Text Parsing

Text parsing is extracting the text and meta data information from the gathered documents. Apache Tika parser is used for this process. It has different parser classes such as HTML parser, PDF parser, Open document parser, Text parser, Office parser etc. The first step is detection of file type, then select appropriate parser, finally performs the extraction task. Tika utilized MIME standard to identify the document type. To detect media type, it uses file extension of a document.

4.2.2.2 Language Identification

A language identification task is detecting the language of documents. Since our system supports two languages of documents i.e. Amharic and English. The need of language identification at indexing time is, document analysis (subsequent NLP sub process) are language dependent tasks. The analysis phases are done separately depending on the language of the document. N-gram language identification algorithm is selected as it is the most effective language identification approach.

N-gram language identification detects a language based on language profiles of N-gram frequency developed for each language. This system builds language profiles for the N-grams in particular language by using training data for the language. Words are used mostly in one language than other language. So that we need sample text in each language we want to detect. N-Gram is an N-character slice of a longer string. Typically, one would slice the string into a set of overlapping N-grams. Blanks appended to the beginning and end of strings in order to help with matching the beginning-of-word and end-of-word.

Apache Tika language identification is an open source, that uses N-gram algorithm to detect language of documents. Since this system does not support Amharic language, we customized this system to support Amharic language by building language profile for Amharic language and including it with other language profiles. In order to classify a sample document, the system computes the overall distance measure between the document profile and the language profile for each language using the out-of-place measure and then pick the language which has the smallest difference.

4.2.2.3 Document Analysis

Document analysis drives a token that is appropriate for searching from documents. In most information retrieval system, tokenization, stop word removal, stemming and normalization are basic tasks performed in the analysis phase. In our system, additional analysis process is performed. That is, sentence detection and named entity tagging of documents. Since the system supports two languages, it has Amharic and English document analysis sub components.

1) English document analysis

The English document analysis is responsible for analyzing English documents. The sub process performed in English document analysis are sentence detection, named entity recognition, tokenization, normalization, stop word removal and stemming.

Sentence detection is a task to compute sentence boundaries. This task help reduce erroneous phrase matches as well as provide a means to identify structural relationships between words and phrases and sentences to other sentences. The named-entity recognition system takes a sentence as input. Because of structural relationships between words and phrases, sentence helps this system to identify named-entity classes.

English named entity recognition it is responsible to detect and tag named entity of text for English documents. Named Entity Recognition (NER) identifies and categorize all named entities in a document into predefined classes like person, organization, location, time, and numeral expressions. It is essential for passage retrieval component to detect which passage contains the answer and guides the system to find answers from documents.

We have used an Apache Opennlp tools for sentence detection and named-entity recognition. The reason we select this open source is because of Apache OpenNLP is very fast and it can process bigger text information in comparison with other NLP tools like Stanford CoreNLP [34].

Normalization sub process changes the letters in documents to lower case. This task is also done in query analysis time. It enables the system to be case insensitive while in searching process. We adopted Apache Solr lower case filter algorithm. This algorithm converts all English words to lower case form.

Stop word removal eliminate English words that are frequently found in documents and less important for detecting relevant document. We adopted Apache Solr stop filter and word list.

Stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form.

Apache Solr has analyzer for analyzing of documents in different language. This tool supports a task like tokenizer, lower casing, stop word removal and stemming task. We adopted Apache Solr English analyzer component to analyze English documents.

2) Amharic document analysis

Amharic document analysis has responsibility of analyzing Amharic documents. The sub process performed in Amharic document analysis are sentence detection, named entity recognition, stop word removal, normalization and stemming.

Amharic named entity recognition is responsible for detecting and tagging words in an Amharic document with Amharic named-entity tagger. It is a process of identifying and categorizing all named entities in an Amharic document into predefined classes like person, organization, location, time, and numeral expressions. We developed an Amharic named-entity recognition system. We develop this system using a maximum entropy model which is a better statistical model in NLP linguistic classification. Maximum entropy models offer a clean way to combine diverse pieces of contextual evidence in order to estimate the probability of a certain linguistic class occurring with a certain linguistic context [35]. It is one of a contribution works during this research. We prepared a training data set or annotated document for training classifier model. Once annotating documents is prepared in the specified format, we generate a model for each class of named-entity (person, location,

organization, time). Then using the generated models, we tag named-entity of Amharic web documents. This is vital for pinpointing answers from large content of documents.

Stop word removal eliminates Amharic words that are frequently found in documents and less important for detecting relevant documents. We adopted a list of stop word that are used in Tessema Mindaye and Solomon Atnafu work [30].

Normalization is responsible to normalize different characters of Amharic that are used interchangeably. Some of Amharic words used interchangeably are ሀ, ኻ, ሐ, ሰ, ሠ, ዐ, አ, ኣ, ጸ, and ፀ. Those words are mapped to one respective character class. We adopted Tessema Mindaye and Solomon Atnafu normalizer [30] in our work.

Stemming is reducing inflected (or sometimes derived) words to their word stem, base or root form. The stemming algorithm developed by Alemayehu Nega and Willett is adopted to develop an Amharic stemmer [36]. After analyzing web documents index will be created by using open source tool for indexing called Apache Lucene.

4.2.2.4 Indexing

Indexing is a process of representing a document that represent a document in a better way for fast searching. This component stores each term in a document with necessary information for searching. This necessary information is how much a particular term appear in a document (term frequency) and how the term is rear (inverse document frequency). We use Apache Lucene for indexing documents. Algorithm 1 describes the steps how the crawled documents are indexed.

Algorithm 1: *Pseudo Code of Indexer Component*

Input: *Documents*

output: *Indexed document*

Begin

For each *document in documents*

Text content = Extracted text content

Language = Tika Language Identifier (Text content)

If Language is English

Analyzed text= English analyzer (document) / (sentence detection, named entity tagger, Tokenization, stop word removing, Normalization, stemming) */*

Else if Language is Amharic

Analyzed text= Amharic analyzer (document) / (sentence detection, named entity tagger, Tokenization, stop word removing, Normalization, stemming) */*

Index = Prepare Index (Analyzed text) //for each work computer weight

*Indexed Document = Indexed Document. Add (Index) /*Assign field values with metadata and content*/*

End For

Return *Document Indexed*

End of Algorithm

4.2.3 Question Translation Component

In cross language information retrieval for matching query and documents either query or document must be translated. We select a query translation approach as it is flexible, less tedious, takes less time and it allows for more interactions with the user. Selecting query translation is the most appropriate approach for translation.

Our system accepts both Amharic and English questions. The translation is performed to translate user question. if the user wants to get result from English language document, question should be translated to English language and vice versa. In QA, translation of question is a full text translation task, for which automatic machine translation (MT) tools are the most appropriate. Due to this reason MT is used for translating question to target language.

As we mention in Chapter Two, rule-based MT approach is not appropriate to take for NLP scares resource language like Amharic. Due to this reason we select SMT for translating query as an appropriate approach.

The performance of SMT system relies on the quality, size and coverage of monolingual and parallel corpus. Amharic-English SMT was developed by Eleni Teshome [37]. The system was not effective due to limitation of the corpus size and domain used in the system. So that it is difficult to use for our system. We developed an Amharic-English SMT system. This also another contribution that have done during in our research work. In statistical machine translation system, translation model and language model used to translate text from source to target language. Bilingual corpus is used to train translation model and monolingual corpus used to train target language model. We prepare a large amount of corpus from different domain (bible, history, constitution). MOSES toolkit is selected for translation question in our work. As it is open source, easily customizable and provides full control over translation process [9]. The four major operations performed in developing Moses SMT system are corpus preparation, data cleaning tokenization and training Moses SMT system.

Corpus Preparation

Training data is an essential component in SMT system. Moses and language modeling toolkit are designed to work on plain text format. So that we converted the data to plain text format. In addition, the parallel corpus should be aligned at sentence level and used by translation system to estimate the target language. The steps performed on the corpus are:

- First all the file formats converted to plain text UTF-8 UNIX format using command line iconv.
- Then align a document on paragraph level using word aligner tools that is GIZA ++.
- Finally splitting sentences and aligning the document using word aligner tools, that is GIZA ++.

Data Cleaning and Tokenization

Tokenization divides the input text into units called tokens where each word can be a number or punctuation. SMT system understand tokens as symbols separated by space. Sometimes word concatenation like negation word should be separated. Moses toolkit has a tokenizer component and we tokenize the corpus by using this toolkit.

After corpus text is tokenized, the next task is cleaning the data. Long sentences and empty sentences are removed as they can cause problem with the training pipeline, and obviously mis-aligned sentences are removed. The corpus that is used for train model must be also free from spelling and grammar error. Since it is difficult to get Amharic spell checker and grammar checker. Corpus cleaning is done manually by checking spelling and grammar.

Training Moses SMT Systems

After the data is fully cleaned and tokenized, we train the SMT models. SMT model contains translated and language model. The Moses toolkit allows automatically training Translated and language model using command line. Language model trained from monolingual data set and distinguish good target usage. whereas translation model trained from parallel data set and maps source to target language. Finally, the Moses decoder component takes source text and SMT models as input and able to translate into the target

text. This component run as a server with a given port number. It employs XML remote procedural call to connect with clients. The server receives input text and send the translated text to our system with given port number. Figure 4.2 shows the architecture of question translation component.

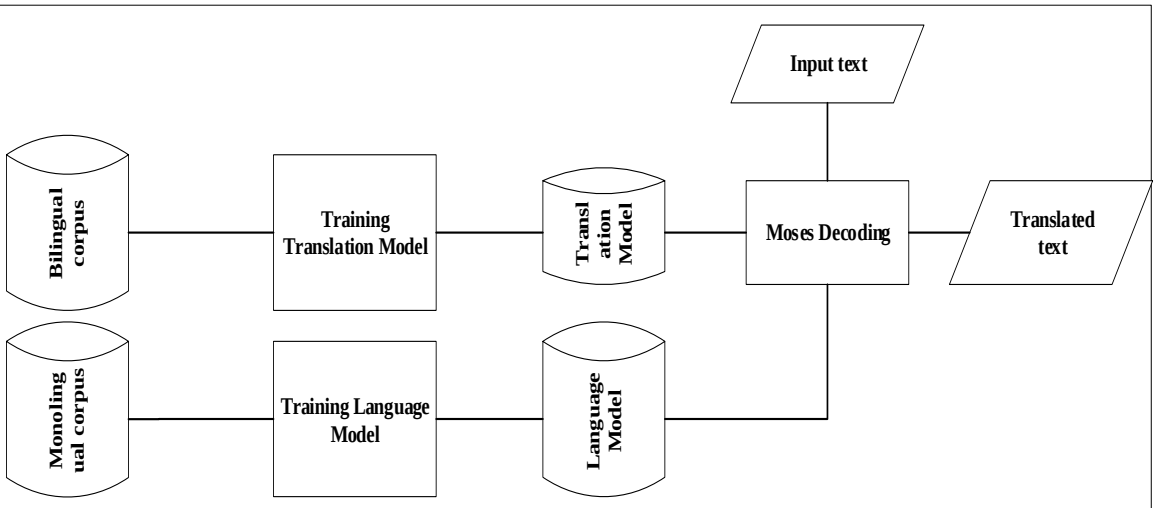


Figure 4.2: Architecture of Question Translation Component

4.2.4 Question Processing

This phase is crucial because the answer extraction strategy completely depends on the information collected from this module. Since our system is capable of accepting questions in both Amharic as well as English languages and question processing is a language specific task, the question processing process has two different question processing modules (Amharic question Processing and English question processing modules).

The question processing includes three sub components. These are question parsing, answer type determination and query generation. Question parsing will parse the given question. The answer type determination component determines the answer type (Person, Location, Organization, Date and Quantity). Query generation has a function of generating query from given question which are relevant words for document retrieval process.

4.2.4.1 Amharic Question Processing

1) Question Parsing

It is responsible for shallow parsing of input question. Shallow parsing (also chunking, "light parsing") is an analysis of a sentence which identifies the constituents (noun groups, verbs, verb groups, etc.). This component enables the system to better understand the subject of the question. We adopted an Amharic-TreebankChunker model of HaBiT system [44] to create shallow parse of the submitted question as it is efficient and well-developed model.

2) Answer Type Determination:

The recognition of expected answer type is the core component in QA system. It guides the retrieval of actual answer by specifying the semantic categories of expected answer. In this module, determining the answer type is done by preparing a pair of question and answer type, extract features from training set, train the system and generating classifier model to determine answer type of user question.

Maximum entropy classification algorithm is selected for generating classifier model as it is a multi-class classifier, since there is multi class or categories (person, place, and organization, time) to determine answer type.

The advantage of the maximum entropy framework is that researcher need only focus their efforts on deciding what features to use and not [35]. This classifier uses the words found in the question as features. Feature selection refers to the process of selecting from the pool of features (which in most cases is very large), a subset which allows the algorithm to generalize. By generalize, we would like the algorithm to perform equally well on some unseen data as it does on the training data. Feature selection methods rank the pool of features using some criteria and then selecting those features above a threshold.

Maximum entropy is an automatic classifier based upon statistical models. To construct a statistical model, first step is to prepare a training data that will be used as a source of knowledge that the classifier uses to categorized unseen objects. The training data contains

sample questions with corresponding to answer type, extract features and then train the data to construct the classifier model.

We extract features of the question using part of speech tagger and chunker. These tools are vital to understand the user query syntactically and semantically. This tool is used to classify a question which has the same meaning and different topology in the same category. Chunking used to find key parts of the given question. The key parts of a question are noun phrase, verb and verb phrase in the question. Table 4.1 shows a question with the key parts that enable to detect the answer type

Features are extracted from each question by using rule based as follow:

- Identifying question word (የት, መቼ, ምን ያህል, ማን) and label it
- Identifying question focus – Noun Phrase which is optional and label it
- Identifying verb next to question word and label it and
- Identifying question support and label it

Table 4.1 : Example for Analyzing Question for Determining of Question Type

Question	ዓመታዊው የፎሊን በዓል የሚከበርበት አመት መቼ ነው?
Question Word	መቼ ነው
Question Focus	አመት
Question Support	ዓመታዊው የፎሊን በዓል
Question Verb	የሚከበርበት

This module takes these features and model to compute answer type of the given question. Feature extraction module takes shallow parse of the submitted question and a set of rules to extract question features. When user entered a question, first the system extracts a feature that helps the classifier to determine answer type. Then the extracted features and model are used for determining the answer type.

3) Query Generation:

Query generation done by query expansion based on the expected answer type. This needs to create a semantically based index of documents based on the expected answer type. This means that a given document's paragraph or even sentence will be analyzed semantically so that it will be labeled as a location name, person name, numerical, etc. as its expected answer type. This module constructs a query for search engine by taking both the parsed query and answer type of the given question. Passing expected answer with query enables to retrieve a set of passages that have answer. In the document retrieval, query matching is efficient to retrieve relevant document while in QA query matching is not sufficient the expected answer may not be found in the retrieved document. In our system to make the system more effective and efficient, the parsed query and answer type are passed to passage retrieval component.

In answer extraction, we have to know where in the document match is found rather than whether the term is found in the document or not. So that we need to do a position-based search. To implement position-based search, creating and storing Term Vectors with position and offset information is necessary. After indexing the documents, searching will be done by using SpanQuery [39]. SpanQuerys allow for nested, positional restrictions when matching documents. It is a Lucene package to enable position-based search. A SpanTermQuery matches all spans containing a particular term. A SpanNearQuery matches spans which occur near one another, and can be used to implement things like phrase search (when constructed from SpanTermQuerys) and inter-phrase proximity (when constructed from other SpanNearQuerys). We construct a Span-NearQuery that contains query term and answer type, because we want to find the specified terms and answer type together.

Finally, we analyze query with query analyzer which is responsible for stop-word removal, normalization and stemming of query. We analyze the user query with the query analyzer for the given field to create SpanTermQuery instances for each term. Algorithm 2 describes the steps how the question is analyzed and Algorithm 3 describes the steps to get key parts (features) of a given question.

Algorithm 2: *Pseudo Code of Question Analysis Component*

Input: *user question*

output: *Spannearquery*

Begin

Parse question = chuker(question)

Question features= feature extractor (parse question)

model =train (training corpus, features)

answer type= answer type determination (model, question features)

Span near query= span query (parse question, question type)

Return *Span near query*

End of Algorithm

Algorithm 3: *Pseudo Code of Feature Extraction Component*

Input: *user question*

output: *a set of features (qw, verb, np, vq)*

Begin

parse-question=part-of-speech tag of each word in a question

qw= select question word(question) /(qw=*የት, መቼ, ምን ያህል, ማን,..*)*/*

verb= find a term part-of-speech tag=verb in question and next to qw

np= part-of-speech tag=noun phrase

vq= verb + question word

features={qw,verb,np,vq}

Return *features*

End of Algorithm

4.2.4.2 English Question Processing

In English question processing, we used the same algorithm with Amharic question processing. But they differ in the NLP tools used for analyzing the input question. As Apache Opennlp has an NLP resource that supports different languages including English, we used this tool in our system for the NLP tasks.

For question type determination, we employ the training data consisting of 1,888 English questions, each hand-labeled by Morton as part of his PhD thesis [40] and Open NLP 's Maximum Entropy classifier for training answer type model, since it is efficient for our work. Open NLP 's Maximum Entropy classifier also takes features and classifier model as an input to classify input question. After we get the parsed query and answer type, then we will generate a span near query by taking both the parsed query and answer type.

4.2.5 Passage Retrieval

The passage Retrieval component is responsible to rank a passage based on term frequency and looking for named entity of answer. Query (Parsed query and named entity) generated from question analysis component will be an input for passage retrieval component. Candidate passage is selected from documents by matching query with search index. A passage that contains both query term and expected answer type is selected as the most relevant passage. For example, if the expected answer type of question is “person” then a passage that does not contain text named entity of “person” in its content are discarded. We used an open source IR system, Apache Solr search engine for retrieving candidate passages.

4.2.6 Answer Extraction

The task performed in answer extraction module is the selection of answers from the set of passages. For a better result, instead of document-based search, position-based search is selected. And to make a position-based, search Lucene’s term vector storage is used. Term vector in Lucene is a data structure that keeps track, per document, of the terms and their frequency and positions within the document. Term vector is truly the final inch for user to reach their search target, the spots where the searched terms exist in the target document. The tasks that are done in this model are:

1) Identify and Scoring Candidate Answer

First identify the start and end position of the query term matches. Then a series of windows of a given number of terms around matching word are constructed. TermVectorMapper is a Lucene class to build a different window around a matching term and these windows are taken as a candidate answer. This approach is selected as it is easy to implement and effective for fact-based QA system [40]. Scoring candidate answer will be done by calculating the weights of windows as follows:

- Calculate weights of the terms, the relative importance of terms in document, are used in computing scores for ranking.
- Score the terms in the main window.

- Score the terms in the window immediately to the left and right of the main window.
- Score the terms in the windows adjacent to the previous and following windows.
- Score any bigrams in the windows. A bonus is given for any bigram matches that is if there's a match of two words in a row.
- The final score for the windows is a combination of all the scores, each weighted separately.

2) Select Top Scoring Answer

After calculating weights of each window separately, ranking will be performed based on the weights of the window. A window which has high weight will be displayed on the top. Candidate answer will be ranked based on their weight.

3) Extract Exact Answer

The exact answer for a factoid questions is a small string. These strings are selected from candidate answer by looking at the entity type. We prepare a set of rules to select a string as an answer. If a question answer type is a person, then select a string named entity type of a person. Algorithm 4 describes the steps to extract an answer for a given question.

Algorithm 4: *Pseudo code of Answer Extraction Component*

Input: *Generated query: query*

Windowsize: 30

Output: *Answers*

Begin

Location of query term =matching (query, document)

*candidate answer = construct window (query term) /*construct 5 windows around query term(main window, first previous window, first follow window, second previous window and second follow window)*

For *windows=windows in window*

For *Wterm=terms in window*

query term weight= calculate term weight (wterm)

End For

bigram weight= bonus+query term weight (give bonus for windows that contains two terms)

weight candidate answer = weight(windows)

return weight

End For

top candidate answer= rank window(weight)

answer=select named entity of answer type (top candidate answer)

Return *answer*

15. ***End of Algorithm***

Chapter Five: Implementation and Experimental Result

This chapter presents the detail implementation of designed for proposed system. First, we discuss the developmental environment that is the tools we utilized for implementing our system. Second, detail implementations of all components which are defined in the architecture will be presented. Finally, the experimental results are presented.

5.1 Development Environment

To implement the designed architecture of cross language QA, several tools are utilized. The system is developed with Intel CORE i5 CPU of 2.3 GHZ speed, a 4 GB RAM and an Ubuntu operating system. Java programing language is selected for writing the code of implementation. It is selected as it is platform-independent, distributed, easy to use, multi-threaded and easily integrated with other system. NetBeans is selected as an integrated development environment (IDE) used in java programming.

Since our system is a web-based QA system it includes components of web search engine like crawler, indexer and query engine. Nutch is an open source crawler that we used to download web documents. Apache Solr is used for analyzing document, indexing and search web document. Moses Decoder is utilized for implementing statistical machine translation to translate the user question.

5.2 Prototype

Figure 5.1 shows the system interface of Amharic-English cross language question answering system.

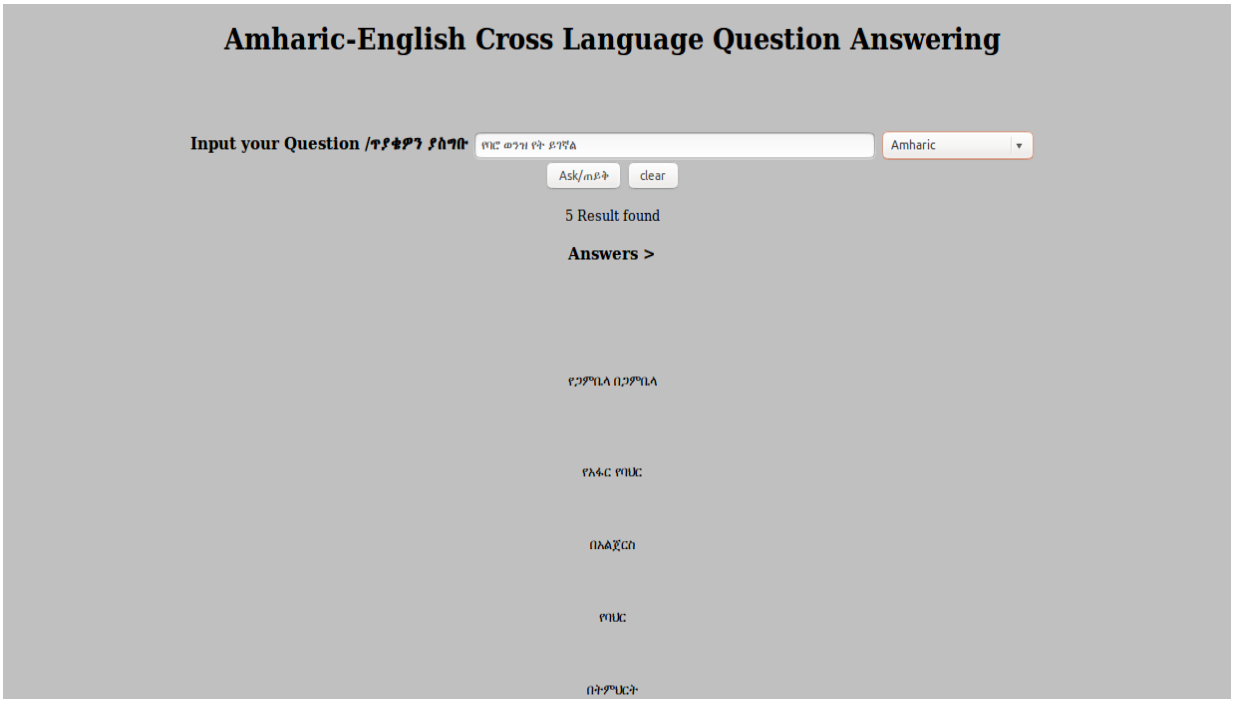


Figure 5.1: Interface for Amharic-English Cross Language Question Answering System

As the system is cross language QA system, it enables retrieving an answer from different language document. Figure 5.2 and Figure 5.3 Shows asking a question of the same meaning in different language (Amharic and English) and getting the same result.

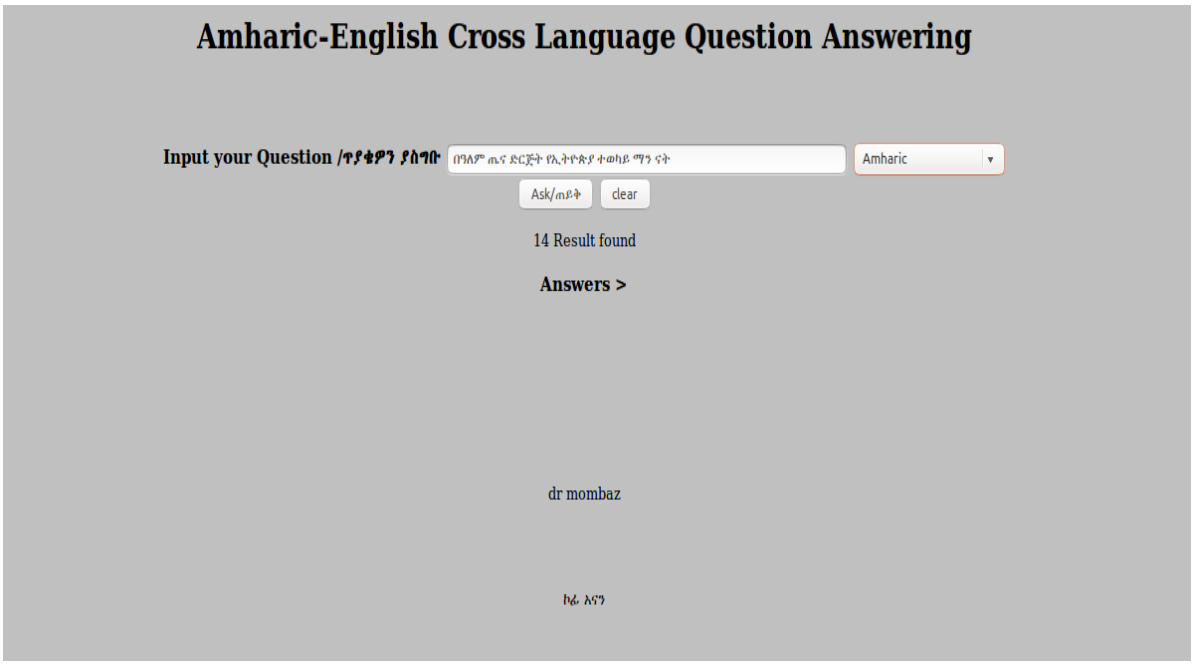


Figure 5.2: Amharic-English Cross Language Question Answering System asking in Amharic

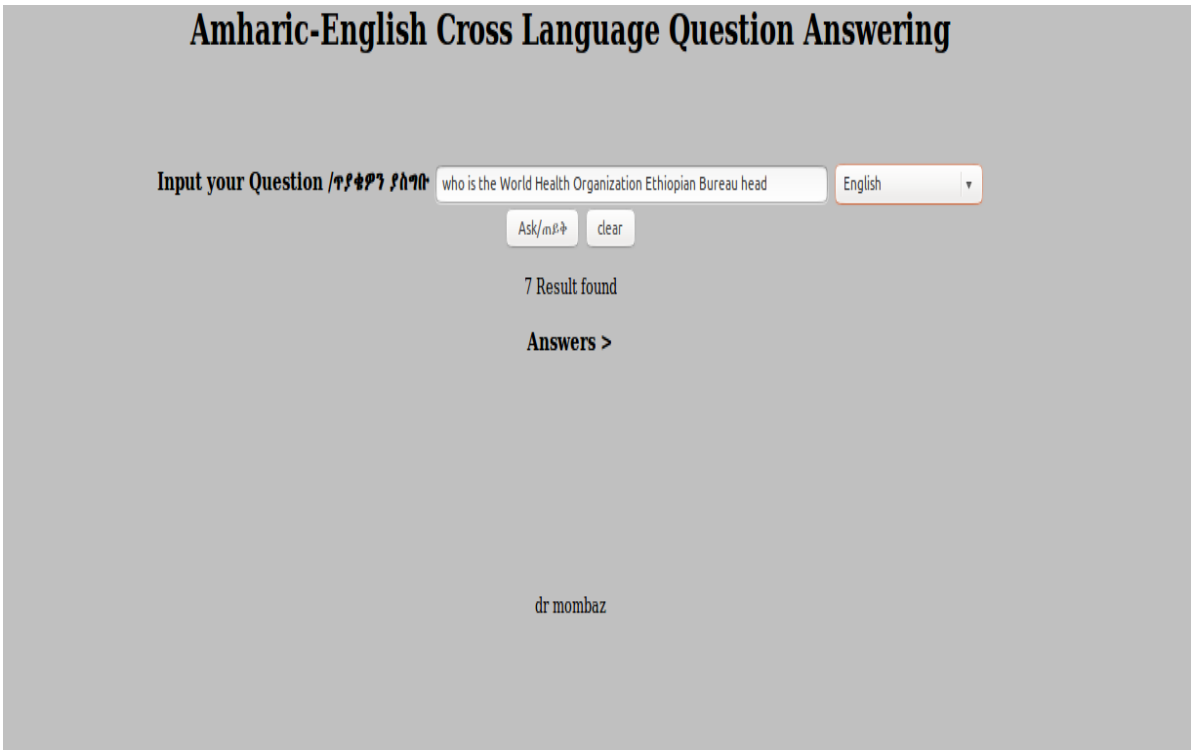


Figure 5.3: Amharic-English Cross Language Question Answering System asking in English

5.3 Experimental Results

We performed an experiment to evaluate effectiveness of the proposed system. To evaluate the effectiveness of our Amharic-English cross-language QA system, precision and recall measures are conducted.

To evaluate the prototype, we prepared a set of questions for the two languages. i.e. Amharic and English. The set of questions, 15 for Amharic 15 for English, from the crawled documents and their answers are manually extracted.

Evaluation of the Overall System

The QA system effectiveness is evaluated by considering the system correctness, completeness and exactness. Precision is calculated as the number of correctly answered questions over the total number of answers (correct, wrong, and No Answer). For a given question q it is computed as:

$$Precision = \frac{Number\ of\ correctly\ answered}{Total\ number of answers}$$

The recall is also calculated as the number of correctly answered questions among the list of expected answer sets where documents will be first analyzed for the presence of correct answers. It is computed as:

$$Recall = \frac{Number\ of\ correctly\ answered}{Total\ number\ of\ expected}$$

1) Monolingual evaluation

monolingual evaluation is done to evaluate how much the system is effective while question and document language are the same language. Amharic and English monolingual evaluation is conducted. To evaluate Amharic monolingual evaluation 15 Amharic questions are input to the system and also to evaluate English monolingual evaluation 15 English questions are input to the system. Table 5.1 shows the experimental results of the monolingual question answering.

Table 5.1: Evaluation result of Amharic and English monolingual question answering

Question	**Precision**	**Recall**
Amharic monolingual retrieval	77%	80.1%
English monolingual retrieval	74%	81.8%

2) Cross-language evaluation

Cross-language evaluation is done to evaluate how much the system is effective while the question and document language are in different language. First translating of prepared question set is given to a linguist (foreign language department post graduate student) who is fluent in both Amharic and English languages and manually translated question set (from Amharic question to English and vice versa). Then the system is evaluated by cross lingual run. Table 5.2 shows the experimental results of the cross-language question answering.

Table 5.2: Evaluation result of Amharic-English cross-language question answering

Question	**Precision**	**Recall**
Amharic-English cross-language retrieval	72%	79%
English-Amharic cross-language retrieval	65%	70%

The experimental result shows that the performance of cross-language retrieval is reduced in comparing with monolingual retrieval. This comes from the translation step tends to cause a reduction in cross-language retrieval performance as compared to monolingual information retrieval.

Chapter Six: Conclusion and Future Works

6.1 Conclusion

Most of web documents are found in the English language. But local QA systems are limited to query only local language documents. Cross-language QA systems enable the user to query from different language document. Developing such kind of a system is essential to access documents without language limitation. Currently, there is no Amharic-English cross language QA system.

We have presented a statistical machine translation approach to translate question language to document language with the purpose of filling language barriers in cross-language retrieval. The translation step tends to cause a reduction in cross-language retrieval performance as compared to monolingual information retrieval. Different approaches for translation are explained.

We make an effort to increase the system effectiveness by implementing semantic indexing or named entity tagging of documents at index time to increase precision of the system. In addition, we make an automatic and effective question type determination by using machine learning and extracting features using rules. Experiments have been performed to assure how much the system is effective. The system is evaluated from monolingual and cross lingual perceptive. Obviously answer retrieving from different languages is less effective as compared to retrieving from the same language due to the efficiency of the translation system used. This translation step tends to cause a reduction in the performance of cross-language retrieval as compared to monolingual information retrieval. The experimental result obtained shows that for Amharic-English cross lingual retrieval we got a result of 72% of precision and recall of 79% and for English-Amharic cross lingual retrieval we got a result of 65% of precision and recall of 70%.

6.2 Contribution of the Work

The main contribution of this thesis work is:

- Designing a general architecture of Amharic-English cross-language question answering, which is developed by creation of question analysis, Amharic named entity tagger, enhancing of Amharic-English Statistical machine translation system as well as adopting of web crawler, document analysis (tokenizer, stop word removal, normalization and stemming) of existing works.

6.3 Future Works

We make an effort to make our system more effective. However, a full QA system is a complex task that needs a lot of time and more effort. It is recommended that this research work can further be enhanced by adding the following features.

- Developing a full QA system that includes non-factoid QA or extending to other question types.
- Integrating syntactic information of source and target language to SMT system to increase effectiveness of the SMT system.
- Integrating an Amharic Word Net with the system.
- Integrating an Amharic spell checker with the system.
- Developing a voice-based QA that enables blind user to use the system.
- Developing a mobile based QA system.

References

[1] Question answering reterived from https://en.wikipedia.org/wiki/Question_answering, accesed on: March 2018 .

[2] B. Katz, G. C. Borchardt, and S. Felshin, “Natural Language Annotations for Question Answering,” *FLAIRS Conf.*, pp. 303--306, 2006.

[3] Seid Muhie, and Mulugeta Libsie. "Amharic Question Answering (AQA)." *Information Foraging Lab*: 98.

[4] Desalgne Abebaw, “LETEYEQ (ሌጢየቕ)-A Web Based Amharic Question Answering System for Factoid Questions Using Machine Learning Approach,” , Unpublished Masters Thesis, Department of Computer Science, Addis Ababa University, March, 2013.

[5] TextREtrieval Conference reterived from https://en.wikipedia.org/wiki/Text_Retrieval_Conference, accesed on: Febrary 2018 .

[6] Mequannint Munye, and Solomon Atnafu. "Amharic-English bilingual web search engine." *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*. ACM, 2012.

[7] I. Hideki, K. Sudoh, and H. Tsukada. "NTT's Japanese-English Cross-Language Question Answering System." *NTCIR*. 2005.

[8] G. Neumann and B. Sacaleanu "A cross–language question/answering–system for german and english." *Workshop of the Cross-Language Evaluation Forum for European Languages*. Springer, Berlin, Heidelberg, 2003.

[9] K. Philipp. "Moses, statistical machine translation system, user manual and code guide." (2014).

[10] H. L. T.-F. B. Kessler, “IRST Language Modeling Toolkit,” vol. 2011, no. 2011/July/02, pp. 1–8, 2011.

[11] O. G. Erik Hatcher, *Lucene in Action, First Edition*. 2005.

[12] N. H. Azim, A. Subki, and Z. N. B. Yusof, *Solr in Action*, vol. 14, no. 1. 2018.

[13] S. Detection and S. Tagging, "Apache OpenNLP Tutorial Apache OpenNLP Tutorial – APIs Named Entity Recognition (NER)."

[14] C. A. Mattmann and J. L. Zitting, *Tika In Action*. 2011.

[15] A. Singhal, "Modern information retrieval: A brief overview," *IEEE Data Eng. Bull.*, pp. 1–9, 2001.

[16] W. B. Croft, M. Sanderson, and T. Strohman, "Search Engines and Information Retrieval," *Search Engine Inf. Retr. Pract.*, pp. 1–9, 2015.

[17] B. Bilotti, Matthew W., Boris Katz, and Jimmy Lin. "What works better for question answering: Stemming or morphological query expansion." *Proceedings of the information retrieval for question answering (IR4QA) workshop at SIGIR*. Vol. 2004. No. 1.2. 2004.

[18] Kibrom Haftu, "Tigrigna Question Answering System for Factoid Questions" , Unpublished Masters Thesis, Department of Computer Science, Addis Ababa University, 2016.

[19] R. Mihalcea and D. I. Moldovan, "Document Indexing using Named Entities," *Text*, vol. 10, pp. 21–27, 2001.

[20] T. Strzalkowski, and S. Harabagiu, eds. *Advances in open domain question answering*. Vol. 32. Springer Science & Business Media, 2006.

[21] J. Lin *et al.*, "What makes a good answer? The role of context in question answering," *Human-computer Interact. INTERACT'03; IFIP TC13 Int. Conf. Human-Computer Interact. 1st-5th Sept. 2003, Zurich, Switz.*, no. September, p. 25, 2003.

[22] J. Prager, "Open-Domain Question–Answering," *Found. Trends® Inf. Retr.*, vol. 1, no. 2, pp. 91–231, 2006.

[23] J. Daniel and J. H. Martin. *Speech and language processing*. Vol. 3. London: Pearson, 2014.

[24] B. F. Green, A. K. Wolf, C. Chomsky, and K. Laughery, “An automatic question answerer,” *West. Jt. Comput. Conf. 19*, pp. 219–224, 1986.

[25] J. Y. Nie, *Cross-Language Information Retrieval One liner Lectures Chapter in Title*. 2010.

[26] D. W. Oard and P. Hackett, “Document Translation for Cross-Language Text Retrieval at the University of Maryland 1 Introduction 2 Cross-Language Information Retrieval,” pp. 1–10, 2007.

[27] N. Katris, R. F. E. Sutcliffe, and T. Kalamboukis, “Using a Cross-Language Information Retrieval System based on OHSUMED to Evaluate the Moses and KantanMT Statistical Machine Translation Systems.,” *Lrec*, pp. 368–372, 2016.

[28] S., “Open Domain Question Answering System Using Semantic Role Labeling,” *Int. J. Res. Eng. Technol.*, vol. 03, no. 27, pp. 104–107, 2015.

[29] L. Augusto, Pizzato, and M. Diego. "Indexing on semantic roles for question answering." *Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering*. Association for Computational Linguistics, 2008.

[30] Tessema, Mindaye, and Solomon Atnafu. "Design and Implementation of Amharic Search Engine." *2009 Fifth International Conference on Signal Image Technology and Internet Based Systems*. IEEE, 2009.

[31] A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke, and S. Raghavan. “Searching the Web”.ACM Transactions on Internet Technology (TOIT), 2001. [32] H. Kazawa, H. Isozaki, and E. Maeda, “NTT Question Answering System in TREC 2001,” *Science (80-.).*, pp. 1–8, 2001.

[32] H. Kazawa, H. Isozaki, and E. Maeda, “NTT Question Answering System in TREC 2001,” *Science (80-.).*, pp. 1–8, 2001.

[33] H. Chedid. *CINDI_QA: a template-based bilingual question answering system*. Diss. Concordia University, 2008.

[34] I. Karlin, “An Evaluation of NLP Toolkits for Information Quality Assessment,” 2012.

[35] A. Ratnaparkhi, “A Simple Introduction to Maximum Entropy Models for Natural Language Processing,” *IRCS Tech. Reports Ser.*, no. May, p. 81, 1997.

[36] N. Alemayehu and P. Willett, “The effectiveness of stemming for information retrieval in Amharic,” *Program*, vol. 37, no. 4, pp. 254–259, 2003.

[37] Elleni Teshome, “Bidirectional English-Amharic Machine Translation: An Experiment using Constrained Corpus”, Unpublished Masters Thesis, Department of Computer Science, Addis Ababa University, no. March, 2013.

[38] K. Nigam, J. Lafferty, and A. Mccallum, “Using Maximum Entropy for Text Classification,” *IJCAI-99 Work. Mach. Learn. Inf. Filter.*, pp. 61–67, 1999.

[39] Accessing words around a positional match in Lucene, accesed from https://lucidworks.com/post/accessing-words-around-a-positional-matc-in-lucene, last accessed on: June 15 2018.

[40] Ingersoll, Grant S., Thomas S. Morton, and Andrew L. Farris. *Taming text: how to find, organize, and manipulate it*. Manning Publications Co., 2013.

[41] Z. Laliwala, *Web Crawling and Data Mining with Apache Nutch*. 2013.

[42] G. Wentzel, “Funkenlinien im Röntgenspektrum,” *Ann. Phys.*, vol. 371, no. 23, pp. 437–461, 1922.

[43] Maybury, Mark. "New directions in question answering." *Advances in open domain question answering*. Springer, Dordrecht, 2008. 533-558.

[44] Amharic Part of Speech tagger, accessed from https://habit project.eu/wiki/HabitSystemFinal, last access on November 20 2018.

ANNEXES

Annex A: Sample Amharic Corpus for Training a SMT model

ጠ / ሚ መለስ ዜናዊ ጦርነት ኳስ ጨዋታ እይደለም ! አሉ
ሰሞኑን በሕወሓት / ኢሕአዴግ ግምገማ ውስጥ ዋነኛው የግምገማ በትር ያረፈው በጠ / ሚ መለስ ዜናዊ ላይ መሆኑ ተደጋግሞ እየተሰማ ነው ።
ከዚሁ ጋር ተያይዞ የጠ / ሚንስትሩ ጋርዶች በሌሎች መቀየራቸው ፣ ከእቶ መለስ ዜናዊ ጋር የሚያገናኙ የቤተ መንግሥት የስልክ ግንኙነቶች መቋረጣቸው በሰፊው እየተነገረ ሲሆን ፣ ማንኛውንም የወቅቱ
ጉዳይ እስመልክቶ መነጋገር የሚቻለው ከውጭ ጉዳይ ሚ / ር መ / ቤት ጋር መሆኑ ታውቋል ።
እቶ መለስ ዜናዊ በዚህ ሁኔታ ውስጥ መሆናቸው እየተነገረ ባለበት ወቅት ነው ከአሜሪካ ራዲዮ ጋር ቃለ ምልልስ ያደረጉት ።
በቃለ ምልልሱም ወቅት በሕወሓት አመራር አካል የኤርትራን ጥያቄ ልዩነት ነበር ይባላል ።
ለዘብተኞች እና ወግ አጥባቂዎች ለኤርትራ መንግሥት በሚሰጠው ልዩ ግንኙነት (ስፔሻል ፈቭር) የተነሳ የተካረሩ ልዩነቶች ነበሩ ይባላል ።
ይህ ምን ያህል እውነት ነው ? ለሚለው ጥያቄ ፣ በተለይም በሕወሓት አመራር አካላት ውስጥ ስለተፈጠረው ልዩነት በኢህአዴግ ውስጥም ሆነ በሕወሓት ላይ የአመለካከት ለውጦች አልነበሩም ፣ ሊኖሩም
እይችሉም በማለት የሚሰነዘረውን የግምገማ ሃሳብ ከሞላ ጎደል ውድቅ አድርጐታል ።
እቶ መለስ ከፖ / ት ኢሣያስ አፈወርቂ ጋር ስላላቸው ግንኙነት ተጠይቀው ሲመልሱ ... ደስ የማይሉ እንዳንድ ባህርያት የማይበትና የምገዛዙበበት ሁኔታ የነበረ ቢሆንም ... የአብሪትና በህዝብ እድልና
በህዝብ ጥቅም ላይ ቁማር የመጫወት እርምጃ ይወስዳሉ የሚል ግምት አልነበረኝም ።
በርግጥ የእምቢተኝነት ምልክቶች ትላንት የመጡ ናቸውም ማለት አይቻልም ።
የቆዩ ናቸው ።
ሲሉ የመለሱ ሲሆን ፣ የተቃዋሚ ድርጅቶችንም እስመልክተው በኢትዮጵያ ሉአላዊነት ፣ በህዝቦቿ መብት እና ጥቅም ላይ የተቃጣ እርምጃ በሚታይበት ወቅት የፖለቲካ አቋምና መስመር ምንም ቢሆን ም
ሁሉም ኢትዮጵያዊ የሃገሪቱን ሉአላዊነት በማስከበር ዙሪያ ተባብሮ መስራቱ የነበረና መሆንም ያለበት ነው ።
የፖለቲካ ልዩነቶቻችን በእንድ ቀን ተስማምተን የሚጠፋ አይደለም ።
የዲሞክራሲ አንዱ ገፅታና ባህርይ የተለያዩ አቋሞች መራመዳቸው ነው ... በማለት መልሰዋል ።
በአሁኑ ወቅት ስለተፈጠረውም ሁኔታ ጦርነት የእግር ኳስ ጨዋታ አለመሆኑን ገልፀው ፣ ይሄ አስቀድሞ እንድ ጐ አባባ ፣ ቶሎ ብላችሁ ጐል እናንተም እግቡ ።
የሚባል ጨዋታ ውስጥ አንገባም ።
ብለዋል ።
ከዚህ አጠቃላይ ሁኔታ መንግስትዎ ምን ትምህርት ወስዷል ?
ለሚለው የመጨረሻም ጥያቄ ፣ ቀደም ሲል በኤርትራ ስውር ትዕቢትና አብሪት እንዳለ የሚታወቅ ነገር ቢሆንም ፣ በኤርትራ ህዝብ ላይ ቁማር ይጫወታል የሚል ግምት አልነበረንም ።
አብሪቱ ይህን ያህል ደረጃ ያመራል ብዬ ግምት አልነበረኝም ።
አሁን የተማርኩት የተገዛዙበኩት ነገር ቢኖር የኤርትራ አመራር አብሪት መጠንና ስፋቱ ማንም ከመገተው በላይ ሆኖ በህዝብ ጥቅም ላይ ቁማር መጫወትን በሚያካትት መልኩ ሊገለፅ የሚቸል ጉዳይ መሆኑ
ተምረናል ፣ ይህን አይተናል ።
ከእንግዲህ ወዲያ ከበስተጀርባችን አብሪተኞች ብቻ ሳይሆኑ በህዝብ እድል ላይ ቁማር የሚጫወቱ ሰዎች እንዳሉ ምንግዜም የምናስታውሰው ነው በማለት ሃሳባቸውን አጠቃለዋል ።
ከፊያሜታ ሰኔ 17 ቀን 1990
የፊእያሚታ መልዕክት
በበሉበት የሚጮኹ ሰሎሞች !
ዛሬ ለደረስንበት የእርስ በርስ ጦርነት መንስኤው የሻዕቢያ እምባገነናዊነት ቢሆንም ፣ ለዚህ ድፍረት የተሞላበት አብሪት ያበቃን ደግሞ በእንድም ሆነ በሌላ መንገድ የወቅቱ አስተዳዳሪ መሆኑ አሌ አይባልም
ኤርትራ ናፅነት ካገኘች በኋላ ህዝቧ ከኢትዮጵያ እንዲገነጠል አስፈላጊው ሁሉ ሲደረግ ነበር ።
ከረፈረንደሙ በኋላ የኢትዮጵያ መንግሥት ጋዜጠኞች የመገንጠል ነጋሪት ሲጐስሙ ተደምጧል ።
የኢትዮጵያ ራዲዮና ቴሌቪዥን የኤርትራ መሪዎች አፈቀላጤ ሆነው እደንቋሪ ፕሮፓጋንዳቸውን ያሰሙን እንደነበር ማንም የሚዘነጋው አይደለም ።
ሌላው ቢቀር ወገኖቻችን ከኤርትራ ተዘርፈው ሲባረሩ ምንም እንዳልተፈፀመባቸው ሱበከውናል ።

Annex B: Sample English Corpus for Training a SMT model

Prime Minister Meles said , War is not a Soccer Game !
Recently , it is often heard that the main stick of criticism rested upon Prime Minister Meles during the TPLF / EPRDF evaluation session .
Meanwhile , it 's widely heard that Prime Minister 's private guards were changed by others , direct telephone lines to the Prime Minister were disconnected in the palace , and one can discuss about anything related to the present condition with the office of Ministry o Foreign Affairs .
It 's in the midst of this situation that Ato Meles gave an interview to voice of America .
During the interview there was a different position about Eritrean question among the TPLF leaders .
Due to the special favor the moderate and the conservatives gave to Eritrean government there was a great difference , it was said .
Responding to whether there was a significant difference between bodies of TPIF EPRDF with regards to the special relationship (special fav) Eritrean government receives he denied that there has been no difference in both TPLF as well as EPRDF , and will never be , and more or less ruled out the issue of evaluation .
Asked about his relationship with president Isayas Afeworkihe responded ... even though there were times when I observe unlikable behavior ... I have never thought he would take a decision merely made out of a game of chance with issues that affect the chances and benef of people .
Actually , signs of obstinacy can 't be said to have come yesterday .
They were there long ago .
And with regards to opposition parties , he answered , At the time when measures are taken against the sovereignty of Ethiopia , her people ' rights and benefits , every one cooperatively works in defense , and this was a common practice and it should be .
Our political difference cannot be resolved in a day 's time .
One of the characteristics of democracy is the interplay of various kinds of positions .
And with regards to the present situation , he stated that war is not a soccer game , we are not going to take part in such a game as ' thi one scored a goal before , so you , too , should score ' kind-of-game .
we won 't participate in such a game .
He said .
When he responded to the last question , what lesson did your government learn from this situation ?
He said although it is known that there 's a concealed arrogance and conceit in Eritrea , never have we expected that he would gamble on Eritrean people .
I never expected this arrogance and conceit would reach as far as this level .
What I have learnt now is that the scope of arrogance and conceit of Eritrean leadership could be revealed in terms of gambling with people 's benefit.This we saw .
From now on , it is something we will always remember that there are not only conceited individuals behind us , but also individuals who gamble on people 's benefit , concluded his speech .
From Fiameta June 24 , 1998
Message of Fiamet .
Dogs that Bark wherever they eat !
Although it is EPLF 's demagoguery the cause for today 's civil war , the reason for entertaining such bold arrogance on us , in o way or another , is undoubtedly the present administrator ..
After Eritrea got freedom all the necessary measures were being taken to secede her people from Ethiopia ..

Annex C: Sample Corpus For Training Amharic Named Entity Model

ፓርላማ አለመግባት አንድ አማራጭ ነው» ቅንጅትና ጎብረት የኢትዮጵያ ዴሞክራሲያዊ ኃይሎች ጎብረት ተቀዳሚ ም /ሊቀመንበር ዶ /ር <START:person> በየነ ጴጥሮስ <END> ተቃዋሚዎች ፓር
ይግቡ ወይስ አይግቡ ለሚለው ጥያቄ ጎብረቱ ሦስት አማራጮችን ይዞ ወደ ምርጫው መግባቱን ገልጸው ከነዚህ አማራጮች ውጭ ፓርላማ አለመግባት እንደ አንድ የትግል ዘዴ ከታመነበት እንደሚቀበሉት
እስታውቀዋል ፡
የቅንጅት ለአንድነትና ለዴሞክራሲ የምርጫ ችግሮች አስወጋጅ ዐቢይ ኮሚቴ ሰብሰበ ኢንጂነር ግዛቸው ሽፈራው በበኩላቸው ቅንጅቱ ፓርላማ እንግባ ወይም እንግባ ብሎ እጀንዳ ይዞ አለመነጋገሩን እው
ይሁንና ነገሩ በጣም አስቸጋሪ ደረጃ ከደረሰ እንደ አማራጭ የምናየው ይሆናል ብለዋል ፡
ሁለቱ የተቃዋሚ ፓርቲ አመራር አባላት በተለይ ከጠቢያ ሪፖርተሮች ጋር ባደረጉት ቃለ መጠየቅ ምልልስ እንደገለጹት ፓርላማ ያለመግባት ውሳኔ የአነርሱ ሳይሆን የመረጣቸው ሕዝብ ነው ፡
ዶ/ ር በየነ ጴጥሮስ ጎብረቱ የያዛቸውንን ሦስት አማራጮች ሲያብራሩ እንደኛ ሙሉ በሙሉ እሸንፎ ፓርላማ መግባት ፣ ሁለተኛ በጣምራ ሥልጣን መያዝ ፣ ሦስተኛ ሙሉ በሙሉ ባያሸንፍ በተቃዋሚነት
ፓርላማ ገብቶ የትግሉን ደረጃ ከፍ ማድረግ ነው ፡
ይሁንና እነዚህ ነገሮች ሁሉ ባይሳካና የህገሪቱ ተጨባጭ ሁኔታ ካስገደደ ፓርላማ አለመግባት እንደሚቻልና ይኼ አሠራርም በአፍሪካም ሆነ በሌሎችም ሀገራት የተለመደ መሆኑን ገልጸዋል ፡
በእሁኑ ወቅት ጎብረቱ ያመነባቸውን የምርጫ ውጤቶች መቀበሉንና ያላመናቸውን ደግሞ አልቀበልም ማለቱን የጠቆሙት ዶ/ ር በየነ ነገር ግን ገዢው ፓርቲ ኢሕአዴግ ጉዳዩን ከሕገ መንግሥት መቀበል
አለመቀበል ጋር እያምታታ እንደሚገኝ ተናግረዋል ፡
በቅርቡ ከጠ/ ሚኒስትር መለስ ጋር ስላደረጉት ውይይት እንደተናገሩት የውይይቱ ዋነኛ ርዕስ የመገናኛ ብዙኃንን አጠቃቀም በተመለከተ ቀደም ሲል ከኢህአዴግ ተወካዮች ጋር ሲደረግ የቆየ ቀጣይ ውይይ
እንደነበር እውስተው ነገር ግን በእውሮፓጎብረት ልዑክ አማካይነት ጠ/ ሚ/ ር መለስን ማነጋገር እንደሚቻል በተደረገ ድንገተኛ ቀጠሮ ውይይቱ መካሄዱንና በመሀሉ ደግሞ ከምርጫው ውጤት ጋር በተ
ሰፋ ያለ ጉዳይ እንደተነሳ አብራርተዋል ፡
ዶ/ ር <START:person> በየነ <END> በኢትዮጵያ ከምርጫው ጋር በተያያዘ አንዳንች ችግር ቢፈጠር ሦስቱ የምርጫ ቦርድ ኃላፊዎች ተጠያቂ እንደሚሆኑና ብቃት የሌላቸው ሰዎች መሆናቸውን ከጠ
መጽሔት ጋር ባደረጉት ቃለ መጠይቅ ምልልስ ገልጸዋል ፡
በተመሳሰይ ሁኔታ የቅንጅት የምርጫ ችግሮች አስወጋጅ ዓቢይ ኮሚቴ ሰብሰቢ ኢንጂነር ግዛቸው ሽፈራው እንደተናገሩት ፓርላማ መግባት ወይም አለመግባት ከተነሰብት ትልቅ ዓላማ አንፃር ምንም እንዳ
ይልቁንም ፓርላማው የሕዝብን መብት ያስከበረ የሕዝብ ፓርላማ ነወይ የሚለው ጥያቄ ምላሽ እንደሚገባው ተናግረዋል ፡
ኢንጂነር ግዛቸው ጨምረው እንደገለጹት ፓርላማ መግባትን በተመለከተ ለቅንጅቱ ተደጋጋሚ ጥያቄዎች እንደሚቀርቡለት አውስተውይሁንና ነገሮች ወዳልተፈለገ አቅጣጫ ካመሩ በጉዳዩ ላይ ከመራጩ
ጋር ውይይት እንደሚያስፈልገው አስረድተዋል ፡
የምርጫ ማጣራቱን ችግሮች በተመለከተ ሰፋ ያለ ማብራሪያ የሰጡት ኢንጂነር ግዛቸው ከመነሻው ጀምሮ ምርጫ ቦርድ የተከተለው አሠራር ገዢውን ፓርቲ ኢሕአዴግን በሚደግፍ መልኩ መሆኑን ጠቁመ
በቅርቡ ጠ/ ሚ/ ር መለስ ለቢቢሲ ጋዜጠኛ እንደገለጹለት ምርጫው ቢደገም ተቀባይነት እንዳለውና ይኼ የሚሆነው ግን አሁን ያሉት የምርጫ ቦርድ ኃላፊዎች በሌሎች ገለልተኛ ወገኖች ሲተኩ ብቻ መ
እስምረውበታል ፡
ከቃሊቲና ከእ አ ማረሚያ ቤቶች 20 የጦር መኮንኖችና ሲቪሊል ታሳሪዎች ተለቀቁ ከ1983 የመንግሥት ለውጥ ጀምሮ በቀይ ሽብርና በዘር ማጥፋት ወንጀል ተሳትፋችኋል በመባል በልዩ ዐቃቤ ህግ ትዕዛዝ ላ
ዘጠኝ ዓመታት ክስ ሳይመሰረትባቸው በቃሊቲና በአዲስ አበባ ማረሚያ ቤቶች በእስር ላይ ከሚገኙት በርካታ የቀድሞ መንግሥት ሲቪሊል ባለሥልጣናትና ከፍተኛ መኮንኖች መካከል ሰሞኑን ሃያ የሚሆኑት
መለቀቃቸውን ለጉዳዩ ቅርበት ያላቸው ምንጮች ገለጡ ፡
በዚሁ መሠረት ብርጋዴየር ጀኔራል <START:person> አብደላ መሐመድ <END> በቀድሞው ጦር የ114ኛ ክር አዛዥ ፣ ብርጋዴየር ጀኔራል <START:person> ሰሙ ንጉሥ መኮንን <END> የምድር ጦ
ገንዘብና ሂሳብ መምሪያ ኃላፊ ፣ ብርጋዲየር ጀኔራል <START:person> ይልማ ኃይለማርያም <END> ፣ አቶ <START:person> ዘለዓለም ዋቀዩ <END> የአዲስ አበባ ኢሠፓ የርዕዮተ ዓለም ጉዳይ ኃ
አቶ <START:person> ዳሪሞ ኤና <END> የሲዳሞ ክፍል ሀገር ገበሬዎች ማኅበር ሊቀመንበር ፣ ወሮ <START:person> ብዙነሽ ቸርነት <END> ፣ ሻለቃ <START:person> ድረስ ተስፋዬና <END
ሻለቃ <START:person> ታመነ አባተ <END> የሁለተኛው አብዮታዊ ሠራዊት አባላት የነበሩ ፣ እንዲሁም ባሻ <START:person> ዮሴፍ ጋሻው <END> ፣ ቺፍ <START:person> ኮስፋዬ ተፈራ
(የቧሀር ኃይል ደህንነት) ፣ አቶ <START:person> ተስፋዬ ቀንዓ <END> ደህንነት ፣ አቶ <START:person> መለስ ተስፋዬ <END> ካድሬ ፣ አቶ <START:person> አለማየሁ ከበበ <END> የደህ
አባል ፣ በተከሰሰበት ወንጀል በቂ ማስረጃ ስላልቀረበላቸው ባለፈው ሰምዓት የተወሰኑት በዋስ ከፊሉ በነፃ መለቀቃቸው ታውቋል ፡

Annex D: Schema.xml configuration for fields

```
<field name="docid" type="string" indexed="true" stored="true" required="true"/>
<field name="file" type="string" indexed="true" stored="true" />
<field name="doctitle" type="text" indexed="true" stored="true" multiValued="true"
termVectors="true" termPositions="true" termOffsets="true" />
<field name="body" type="text" indexed="true" stored="true" multiValued="true"
termVectors="true" termPositions="true" termOffsets="true" />
<field name="docdate" type="date" indexed="true" stored="true" multiValued="false"/>
<field name="titleBody" type="text" indexed="true" stored="false" multiValued="true"
termVectors="true" termPositions="true" termOffsets="true" />
 <field name="category" type="text" indexed="true" stored="false" multiValued="true"
termVectors="true" termPositions="true" termOffsets="true" />
<field name="content" type="text" indexed="true" stored="true" multiValued="true"
termVectors="true" termPositions="true" termOffsets="true" />
```

Annex E: Schema.xml configuration for query and document analysis

```
<fieldType name="text_en" class="solr.TextField" positionIncrementGap="100"
autoGeneratePhraseQueries="true">
    <analyzer type="index">
            <tokenizer class="QAS.SentenceTokenizerFactory"/>
            <filter class="com.QAS.solr.NameFilterFactory"/>
            <filter class="solr.LowerCaseFilterFactory"/>
            <filter class="solr.SynonymFilterFactory" synonyms="index_synonyms.txt"
ignoreCase="true" expand="false"/>
            <filter class="solr.StopFilterFactory"
             ignoreCase="true"
             words="stopwords.txt"
             enablePositionIncrements="true" />
            <filter class="solr.PorterStemFilterFactory"/>
             </analyzer>
    <analyzer type="query">
            <tokenizer class="solr.WhitespaceTokenizerFactory"/>
            <filter class="solr.StopFilterFactory"
             ignoreCase="true"
             words="stopwords.txt"
             enablePositionIncrements="true" />
           <filter class="solr.WordDelimiterFilterFactory" generateWordParts="1"
generateNumberParts="1" catenateWords="0" catenateNumbers="0" catenateAll="0"
splitOnCaseChange="1"/>
            <filter class="solr.LowerCaseFilterFactory"/>
            <filter class="solr.PorterStemFilterFactory"/>
            </analyzer>
           </fieldType>
```

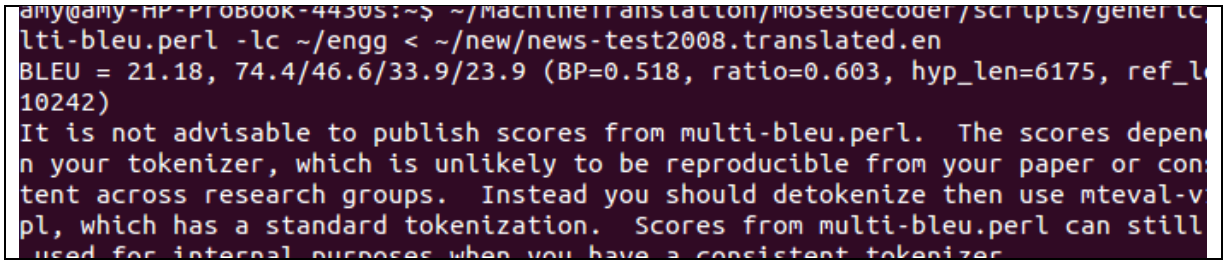
```
<fieldType name="text_amh" class="solr.TextField" positionIncrementGap="100"
autoGeneratePhraseQueries="true">
    <analyzer type="index">
      <tokenizer class="QAS.solr.SentenceTokenizerFactory"/>
      <filter class="QAS.solr.AmharicNamefilterFactory"/>
        <filter class="QAS.texttamer.solr.SampleClient"/>
      <filter class="QAS.solr.StopWordRemovalFactory"/>
      <filter class="QAS.solr.NormalizerFactory"/>
       <filter class="QAS.solr.AmharicStemmerFactory"/
    </analyzer>

    <analyzer type="query">
      <tokenizer class="solr.WhitespaceTokenizerFactory"/>
     <filter class="QAS.solr.StopWordRemovalFactory" />
     <filter class="QAS.solr.NormalizerFactory"/>
     <filter class="QAS.solr.AmharicStemmerFactory"/>
    </analyzer>
   </fieldType>
```

Annex F: Sample Question Sets with Answer distribution Statistics: √ = Correct Answer, WR = Wrong Answer, NA = No Answer

Question	**1st Rank**	**2nd Rank**	**3rd Rank**	**4th Rank**	**5th Rank**
1. የባሮ ወንዝ የት ይገኛል?	√	WR	WR	WR	WR
2. የኢትዮያ ጠቅላይ ሚኒስትር ማን ይባላሉ?	WR	√	WR	WR	WR
3. የኢትዮጵያ ምሁራን ፍለሰት የተጀመረው መቼ ነው?	WR	WR	WR	WR	WR
4. ኢትዮጵያና ኤርትራ ሁለት አመት የወሰደባቸውን ደም አፋሳሽ የድንበር ጦርነት በሰላም ለመጨረስ ላይ የተስማሙት የት ነው?	WR	WR	WR	√	WR
5. የኢትዮጵያ ኦርቶዶክስ ቤተክርስቲያን ስደተኛ ሲኖዶስ ዋና ፀሐፊ ማን ናቸው?	√	WR	WR	WR	WR

Annex G: Amharic-English SMT model testing result



Annex H: Sample Amharic Questions with precision and recall result

Sample Questions	Precision	Recall
1. ለተባበሩት መንግስታት ድርጅት ዋና ፀሀፊ ማን ናቸው?	0.7	0.67
2. የባሮ ወንዝ የት ይገኛል?	0.88	1
3. የኢትዮጵያ ጠቅላይ ሚኒስትር ማን ይባላሉ?	0.75	0.7
4. ኢትዮጵያና ኤርትራ ሁለት አመት የወሰደባቸውን ደም አፋሳሽ የድንበር ጦርነት በሰላም ለመጨረስ ላይ የተስማሙት የት ነው	0.8	0.75
5. የኢትዮጵያ ኦርቶዶክስ ቤተክርስቲያን ስደተኛ ሲኖዶስ ዋና ፀሐፊ ማን ናቸው	0.67	0.67
6. ለተባበሩት መንግስታት ድርጅት ዋና ፀሀፊ ማን ናቸው	0.82	0.8

Annex I: Sample English Questions with precision and recall result

Sample Questions	Precision	Recall
1. Who is the prime minister of Ethiopia?	0.82	0.89
2. Where is the Baro River is found?	0.77	0.79
3. where is Ethiopia and Eritrea agree to end bloody border war?	0.7	0.78
4. Who is the Secretary General for the United Nations?	0.62	0.9
5. when is Ethiopia and Eritrea agree to end two-year bloody border war	0.85	0.85
6. When is the public meeting of the Ethiopian Democratic Party (EDP) held	0.68	0.7