ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Ontology-based Semantic Indexing for Amharic Text in Football Domain

By

Genet Asefa Gesese

Advisor: Fekade Getahun (PhD)

A THESIS SUBMITED TO

THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCINECE IN COMPUTER SCIENCE

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Ontology-based Semantic Indexing for Amharic Text in Football Domain

By

Genet Asefa Gesese

Advisor: Fekade Getahun (PhD)

Signature of the Board of Examiners for Approval

Name	Signature
1. Dr. Fekade Getahun, Advisor	_____
2. _____	_____
3. _____	_____

March, 2013

DEDICATED TO:

Grandma (Abye)

Acknowledgments

Most of all, I would like to thank God, who makes everything possible, for helping me pass all those hard times that I will never forget in my life.

I owe my deepest gratitude to my advisor Dr Fekade Getahun for his time, patience and undeniably helping comments all the way through this study. He really was an inspiration for me to proceed whenever I face difficulties and he is easily approachable. This thesis would not have been possible without his constructive comments on every aspect of the study.

I would like to thank Ato Tesfaye, Health and Physical Education (HPE) expert in Ministry of Education (MOE), for facilitating access to resources from Ethiopian Football Federation (EFF) and for his very helpful expert advises on the main parts of the work.

My appreciation also goes to Ato Getachew Abebe and Zewudinesh Yirdaw, staff members of EFF, for spending their valuable time to respond to my questions and for providing the necessary data. Besides, I am very grateful to “Wondirad Preparatory School” teachers who have helped me by providing necessary materials and significant expert advises on the linguistic part of the work.

I want to express my heartfelt thanks to Mr. Michael Gasser for his unreserved assistance throughout this study. I would also like to express my appreciation to my family members and friends who have helped me in so many ways.

I really like to pass my sincere gratitude to my wonderful classmates for creating such an exciting batch. I will always cherish the time we have spent together. Finally, I want to thank all the people who have contributed in one way or another on this thesis work.

Contents

List of Tables

List of Templates

List of Figures

List of Algorithms

List of Annexes

Acronyms

AI	Artificial Intelligence
CT :	Concept Tagger
CW:	Concept Weighting
DR:	Document Retrieval
EFF:	Ethiopian Football Federation
ERTA:	Ethiopian Radio and Television Agency
IC:	Individual Creator
IDF:	Inverse Document Frequency
IE:	Information Extraction
IR:	Information Retrieval
KR:	Knowledge Representation
LSI:	Latent Semantic Indexing
OP:	Ontology Population
POS:	Part Of Speech
QC:	Query Caching
SVD:	Singular Value Decomposition
TF/IDF:	Term Frequency/Inverse Document Frequency
WIC	Walta Information Center

Abstract

Enormous amount of data has been produced in electronic format in Amharic language which led to information explosion. This has created a major challenge for information managers in processing information and providing it to users quickly and easily. Therefore, some indexing methods have been proposed for Amharic language by researchers so far. However, these methods are not capable enough to capture the semantics of documents. In this research, an effort has been made to build a semantic indexer for Amharic football news articles by applying domain ontology.

The main purpose of the study is to construct an index which is embedded with the ontology so as to minimize query processing time. Ontology development, Document indexing, and Query processing are the core components of the study. Document indexing component is composed of Concept Tagger, Information Extraction, Concept Weighting, and Ontology Population modules. The role of Concept Tagger module is to annotate documents with concepts from the ontology whereas Information Extraction Module is responsible for identifying new individuals and determining the relationship between concepts in the tagged/annotated documents. The Concept Weighting module involves calculating weights for concepts and individuals using the domain ontology. The weights computed for the concepts and individuals are added to the ontology by using the Ontology Population module.

The query processing component is built with the purpose of testing the performance of the indexer with user queries. This component has Query Caching, Individual Creator, Document Retrieval, and Document Ranking modules. Query caching is the process of registering original and tagged query pairs in order to avoid running preprocessing and tagging modules whenever the same query is posed by users. Individual Creator module is intended to produce new individuals from queries and adding them to the ontology. Finally, the Document Retrieval and Document Ranking modules are used to retrieve and rank documents according to their level of relevance. Concept reasoning or inferencing is the main task in the document retrieval process.

The precision, recall, and F-measure techniques are used to evaluate the performance of the proposed system and the classical IR based on the relevance information provided by experts. The result shows that the proposed semantic indexer has better performance than the lucene indexer used in the classical IR.

Key Words: Semantic indexing, Football domain ontology, Rule-based information extraction, Semantic information retrieval, Query processor, Concept tagging.

Chapter One - Introduction

1.1 Overview

Since the digital technology has emerged, massive amount of data has been produced in electronic format. In addition, the data which had been held on paper for long time has been converted to soft copy to share them with the public. As the available electronic data in every sector got increased, accessing valuable information out of them has become a critical issue.

In order to dig up information from huge repositories with less time and energy, we need to use an effective indexing mechanism. Indexing is a way of locating documents using representative terms or concepts to make information searching or document categorizing easy and quick. There are two main categories of indexing; manual and automatic indexing. Manual (Human) indexing is the process of representing documents by domain experts without computerized systems whereas automatic indexing is done with automated systems without any human intervention [1].

Indexing can be used in different applications like information retrieval (IR), Document categorization and so on. In the field of IR, indexer is used by search engines to represent the content of a document with short and content-bearing terms so that the retrieval process can have a great performance. In the case of document categorization, index terms are used to identify in which predefined category a document belongs.

Various researches [2,3,4,5,6,7,8] have been undertaken in the area of document indexing for different languages with an intention of bringing a means for better document processing and retrieval. In general, there are two main categories for these indexing approaches; keyword-based and concept-based indexing (semantic indexing). Keyword-based methods are not capable of capturing the implicit relation among terms or the semantics of the words in the document. To remove this weakness, concept-based indexing comes into existence. The purpose of this thesis

is to come up with a semantic indexing method dedicated to Amharic documents that could show improvement over the existing indexing schemes with better performance and efficiency.

Ontology-based Semantic Indexing for Amharic Text in Football Domain is a research intended to provide semantic indexing approach for football news text. This research is composed of three main components - ontology development, document indexing, and query processing. In the ontology development component, football domain ontology is built with the intention of providing a domain specific knowledge base as an input for the document indexer and query processor components. The document indexing task is the core part of the research in which the indexer is designed and implemented whereas the query processing task is responsible for designing and implementing an information retrieval system which utilizes the proposed indexer.

1.2 Motivations

In organizations like Walta Information Center (WIC) and Ethiopian Radio and Television Agency (ERTA), considerable amount of football news written in Amharic language are available being accumulated in different repositories. The size of these news documents has been increasing dramatically and yet there is a need to share them to the public. To make this happen we need to have a search engine which is responsive for Amharic language. In order to have a quality search engine every component of the engine must be developed very well. Indexing is one of the components in IR which plays a great role in making searching easy, quick and effective. In-line with these two prominent researches has been conducted in the area of Amharic document retrieval. The first one is "Design and Implementation of Amharic Search Engine" [2] and the second one is "Enhanced Design of Amharic Search Engine**"** [9]**.** Both of these search engines utilize a keyword based indexing method which is incapable of representing the semantic of the content of documents.

When undertaking a research that involves developing an automatic document processing on documents written in Amharic language, the language itself has many challenges to be dealt with. In Amharic language lexical variation is very common [10], one word may have more than one meaning which is referred to as *Polysemy*. For example, the word "ለጋ" has two different meanings; it can be interpreted as either "kicking a ball" or "very young". In addition, more than

one word may have similar meaning. For instance, the words መምታት and መለጋት have similar meaning; which is referred to as *synonymy*. One indexing scheme should have the capability of dealing with these characteristics of the language.

The classical indexing mechanism, exact term matching, was not capable of dealing with these two vital properties of the language (Synonymy and Polysemy). In [3], the researcher applied latent semantic indexing (LSI) with Singular Value Decomposition (SVD) method in an attempt to solve the problem of VSM scheme. The researcher had tried to construct a semantic indexer which can be able to consider documents which do not share common words with the query by exploring the LSI method. The LSI extracts concepts from a given corpus by looking at the words that occur together frequently without giving emphasis to the relationship between concepts. However, this approach is incapable of handling indirect queries because it does not consider the relationship among concepts. The other problem of the LSI methodology is providing irrelevant query results out of the user's demand. The other problem of this methodology is as the size of the documents gets enlarged, the performance of the indexer degrades.

Let's examine the following scenario which clearly shows the difference among the automatic indexing methodologies discussed so far. Query 1: "የቅዱስ ጊዮርጊስ አሰልጣኝ ስም". The existing solutions to answer this query are:

- ✓ Mindaye, T.'s work ("Design and Implementation of Amharic Search Engine") [2]: handles the above query using lucene's indexing scheme. It extracts keywords from the query and compares the words in the query with the terms in the index which is called as exact matching. If the query and the index have no word in common, then the system will respond no result to the user (low recall). Besides, any document which has the word "ስም" may be retrieved even if it is irrelevant to the user (low precision).
- ✓ Hailemeskel, T.'s work (LSI) [3]: This method takes the query as a pseudo document and adds it to the vector space which is filled with documents. Then from the vector space, documents which are close to the pseudo document will be retrieved. In this case, if one of the documents which are close to the pseudo document is the one in which the term "ጊዮርጊስ" has occurred frequently, then the document will be retrieved though it is irrelevant (this decreases the precision rate). Furthermore, if there is a document which

talks about coaches (“አሰልጣኞች”) and the threshold value (the dimensionality variable k) is too small, then this document will not be retrieved though it may have additional useful information to the user (lower recall rate). In addition, this method is not capable of handling indirect queries.

Unlike the classical indexing scheme (keyword based indexing), the proposed semantic indexer is intended to provide a concept-based indexer by using concepts as index terms rather than words. It annotates documents with concepts by using football domain ontology and it applies a rule based information extraction technique to identify new concept instances/individuals from documents, i.e., to capture the semantics of the contents of documents. The annotated concepts and the newly extracted individuals are populated into the ontology along with their corresponding document links/URLs. This allows the ontology to be used as both a knowledge base and an index so that any information retrieval (IR) system that utilizes this index doesn’t need to perform query expansion and will be able to respond to most queries with less time. Therefore, the index built using the proposed method contains concepts which are capable of representing the semantics of the documents they are referring to.

An IR system that uses the semantic indexer proposed in this study handles queries in different fashion than the existing systems. Concept reasoning/inferencing is applied on the concepts and individuals extracted from a query using the domain ontology. This allows the system to deal with indirect queries. For instance, while processing the query given above, Query1, query items (concepts and individuals), which are considered as capable enough to represent the content of the query, are extracted using the domain ontology and inferencing will be done on each of them. The items that represent the query are - “ቅዱስ ጊዮርጊስ” and “አሰልጣኝ”. When applying inferencing on these items the term “ሰዉነት ቢሻዉ”, the individual of the concept “አሰልጣኝ”, is inferred and documents that contain this individual are retrieved. Therefore, the proposed system is capable of handling indirect queries by using concept reasoning/inferencing.

1.3 Statement of the Problem

From the few researches conducted in the area of indexing for Amharic text, the only one which aimed to consider the semantics of the documents was the "Amharic Text Retrieval; an Experiment Using Latent Semantic Indexing with Singular Value Decomposition" [3]. But this method has its own limitations which has a considerable impact on its performance. The problems of this latent semantic indexing methodology are its incapability of handling indirect queries, getting out of user demand while retrieving documents, and scalability issues, as mentioned in Section 1.2. The rest of the indexing methods are keyword based methods which do not represent the meaning of the content of the document. Even though there are different researches conducted for other languages in this area, they will not be applied to Amharic text retrieval. Amharic language has its own unique Geez alphabets and different writing system (Ethiopic script) which needs complex Amharic analyzers.

Because of the poor quality of the existing Amharic indexing schemes, as it is discussed in detail in Section 1.2 and Section 3.1, it is desirable to look for a different method of semantic Amharic text indexing. The intention of this research is to come up with a new indexing methodology to be demonstrated on football news documents written in Amharic language. It is expected to eliminate the problems of the existing keyword based and LSI methods by integrating a knowledge base (ontology) with the documents. The football domain is selected because it is easy to get the necessary data to construct ontology and to collect news documents for the testing purpose. Even if the research is done specifically on football domain it can be used for other domains as well with some modification.

Thus the key research questions associated to this thesis work are:

- ✓ How can ontology be used to improve document indexing?
- ✓ How can information extraction concept be applied in the process of document indexing and what advantages can it bring?
- ✓ What is the advantage of ontology inferencing in the document indexing process?
- ✓ What are the differences between the semantic indexing schemes and the keyword based schemes?

1.4 General and Specific Objectives

General Objective

The overall objective of this research is to build an automatic semantic indexer for Amharic football news documents.

Specific Objectives

To accomplish the abovementioned general objective, the following specific objectives are devised.

- ✓ Assess capability of existing text indexing approaches
- ✓ Understand the football domain
- ✓ Propose a semantic knowledge base– Ontology dedicated to the football domain
- ✓ Extract concepts embedded in the news text
- ✓ Provide an approach that extract neighboring concept of a given word
- ✓ Build up an indexing structure by integrating the ontology with the concepts of documents
- ✓ Collect data from Ethiopian football federation and news agencies, populate the ontology with concept instances
- ✓ Test performance of semantic indexer using sample queries; measure the relevance of the approach against existing approaches using precision, recall and expert judgment.

1.5 Justification of the Study

In this very changing world, new technologies have been emerged dramatically and so is the information need of the people. Therefore, almost in every discipline, people are using

automated systems that generate information in electronic format in different natural languages. This has brought a problem of dramatic increase in information and it has become a major issue in the field of information management [11]. However, in order to facilitate accessing huge amount of information, automated systems that use Information Retrieval (IR) technique are needed. Amharic is first language for 27 million people and second language for 7-15 million of people in Ethiopia [12]. As it is the work language of the country, enormous amount of data, written in Amharic, have been produced and made available for use. The existing indexing schemes developed for Amharic language are not capable in providing content based indexing methods on the semantic information embedded in the documents.

Indexing is one of the components of information retrieval systems which require an intensive design for better performance of the retrieval systems. The Amharic language has challenging characteristics like synonymy and polysmy that need handling when the language is processed in a machine. Semantic indexers are supposed to handle these characteristics of the language. This research is aimed to build a semantic indexer by using domain ontologies. The proposed method is intended to bring improvement over the existing Amharic indexing methods so that a better information retrieval system can be built.

1.6 Scope and Limitation of the Study

1.6.1 Scope

This research has been conducted to explore the advantage of knowledge bases, ontologies in particular, in semantic indexing for Amharic texts.

1.6.2 Limitation

- Figures, tables, and images in Amharic news document are not taken into consideration for indexing purpose except the texts written using sequences of characters.
- No word sense disambiguating has been done to identify exact contextual meaning of a word with different meaning i.e. polysemy is not handled in this research.

1.7 Methodology

In order to accomplish the general and specific objectives of this study, different methodologies have been applied.

- ✓ **Literature Review**

Extensive literature reviews will be conducted to acquire enough understanding of the various components of the indexer. Specifically, literatures in the area of automatic indexing (concepts and approaches), ontologies, Amharic language writing system, information extraction methods, and existing automatic indexing researches for Amharic and non-Amharic languages will be reviewed. Reading and criticizing all these literatures and related works helps to gain the required knowledge on a certain subject and also assists in identifying the right methods and tools for implementing the different components of the system. Hence, different research papers, books, and web sites will be exploited.

- ✓ **Data Sources for the Experiment**

Two categories of data are required for the purpose of system testing; data to populate the ontology and data for applying text extraction and indexing. The first category contains a set of data which is a collection of concept instances to be added to the ontology and it will be gathered from the Ethiopian Football Federation (EFF). On the other hand, the second category of data is the collection of football news, which will be collected from WIC information center, to test the indexer performance.

- ✓ **Experimentation method**

In order to accomplish the objectives of the research, different methods and tools will be engaged in the process. The initial ontology is developed using the OWL language on protégé tool[1]. Then it is moved to the eclipse Java[2] environment to manipulate the ontology using Jena framework[3]. Mysql database[4] was used as a backend for Jena to sore the ontology. Because it is easier and takes less time to develop a program in Python than in Java, the Python programming language is used to write the module for extracting structured information from documents. The SPARQL (RDF query language[5] is used to access the concepts in the ontology.

[1] http://protegewiki.stanford.edu/
[2] http://www.eclipse.org/webtools/
[3] http://jena.apache.org/
[4] http://www.mysql.com/
[5] http://www.w3.org/TR/rdf-sparql-query/

- ✓ **Testing and Evaluation**

To evaluate the performance of the proposed semantic indexing method, the method will be tested using 138 news documents and 25 queries. The relevance of the retrieved documents will be checked by experts on the domain and some readers of sport news. Based on the users and the experts' feedback, the result of the test will be evaluated by applying recall, precision and F-measure techniques.

1.8 Application of the Study

Apart from being a research to fulfill the requirement of the Masters program, the result of this study is believed to be used either as an input for other researches or can be put into use in different fields. The possible applications of the indexer which is built with this research are;

- ✓ To be used in Amharic search engine for better performance
- ✓ To be used as input to design semantic news classifier
- ✓ To be used as an input in other information management tasks like text summarization, information extraction and etc.
- ✓ In addition, the study can open a way for further researches in the area of semantic text indexing methods for Amharic language.

1.9 Thesis Organization

The rest of the chapters in this report are organized as follows. Chapter two deals with the literatures reviewed so far. It includes Concepts Related to Automatic Indexing, Approaches to Automatic Indexing, Ontologies, The Amharic Language Writing System, and Information Extraction Methods. Chapter three discusses on the existing indexing methods for Amharic and English Languages. Chapter four describes the Design and Implementation of the proposed semantic indexer. The Experiment and Evaluation of the system is discussed in chapter five. Finally, conclusions drawn from the thesis result, the contributions of this research work and recommendations on possible future works related to this research are given in chapter six.

Chapter Two - Literature Review

In this chapter, extensive reviews of general concepts in automatic indexing, known approaches and methods of indexing, ontologies and available tools for ontology development, relevant components of the Amharic language, and finally review of the well-known information extraction approaches and methods are presented. The extensive literatures have been reviewed to understand the problem associated to the realm of this thesis and also to identify appropriate solution.

2.1 Concepts Related to Automatic Indexing

Indexing is the process of representing documents so as to make searching information out of documents and locating them easily, quickly and effectively. There are two broad categories of indexing, manual and automatic indexing. Manual indexing, also called as human indexing, is one kind of indexing methods where indexing experts, human indexers, use their domain knowledge to understand the contents of the documents and select candidate terms which they think are capable of representing the meaning or concept of the documents [11]. Automatic indexing refers to an indexing mechanism where indexing is done by computer algorithms without the involvement of humans. As it is indicated in [11], manual indexing is slow and expensive whereas automatic indexing is cheaper, fast, and easy to modify.

Automatic indexing can be used in different computer applications like document categorization, information retrieval, information extraction and so on. Indexing is one component in document categorization which has a substantial impact on the performance of the categorizer [13]. Classifying documents into the correct category depends on the performance of the indexer; as the quality of the indexer increases so is the performance of the categorizer. Additionally, indexer is a vital component of information retrieval (IE) systems and its performance has a

considerable impact on the overall system performance. For example, the search engine developed by Hassen [9] uses a lucene indexer to build index terms for documents.

2.2 Approaches to Automatic Indexing

According to the authors in [14], the different indexing approaches are grouped into three major categories depending on the extraction techniques used to extract index terms from documents (i.e., statistical, probabilistic, and linguistic methods). The linguistic approaches have two main subdivisions: semantic and syntactic methods. The details about all of these indexing approaches are discussed in the following sections.

Statistical Approach

In this approach, the terms which are believed to be capable of reflecting the content of the documents are extracted by applying statistical methods on the words that appear in the entire document collections. The basic notion of this approach is to explore the occurrences of concept-bearing words in one document and in the collection as a whole. Inverse Document Frequency (i.e., IDF) is one of the well-known statistical methods. It takes those words which occur in a few documents from the entire collection but appear frequently in a single document as index terms [15]. Term frequency and document frequency needs to be calculated before computing IDF. Term frequency is the number of occurrences for the term in a document, whereas document frequency is the number of occurrences for the term in the whole collection [16]. There are other techniques which belong to this approach like N-gram based method to extract concept-bearing terms from documents [4], and Statistical Corpus-Based Term Extractor [17]. The research conducted for Amharic text indexing using N-gram technique is discussed in detail in Section 3.1.

Probabilistic Approach

Probabilistic techniques are based on the interdependency among terms and the probability that these closely interconnected terms exist in relevant documents to extract more document-bearing

complex index terms [18]. As it is stated in [14], even though the dependencies among terms can be explicitly defined by users, this approach and the statistical approach as well are not likely to give quality index terms because it is not possible to argue that terms that occur together are necessarily related semantically. Probabilistic models of indexing [19] is one of the applications developed using this approach.

Syntactic Approach

The syntactic approach makes use of the sentence structure of the document in order to extract the relationship among words so that it can identify the appropriate index terms to represent the document [20]. Using syntactic information makes this approach able to overcome the problem of generating incorrect phrases in non syntactic approaches. However, this approach is incapable of capturing the semantics of the content of documents. There are some research works conducted using this indexing approach such as The Effect of Syntactic Phrase Indexing on Retrieval Performance for Dutch Texts [21] and Syntactic Approaches to Automatic Book Indexing [15].

Semantic Approach

The semantic approach is more dedicated to capture the meaning of the contents of the documents using concepts rather than words as index terms. Since one concept can be expressed using different terms in different documents, using key words as index terms could not guarantee the appropriateness of the terms in representing the content of a document. This approach looks at the semantic relationship among index terms (concepts) to overcome the problems of the key word based approaches [22]. Semantic methods are expected to handle the synonymy, polysemy and other challenges of natural languages. Latent Semantic Indexing (LSI) is one of the known approaches that identify implicit concepts from documents using mathematical computations [23]. The research conducted for Amharic text using the LSI method [3] is presented in Section 3.1.

According to the authors in [24], there are two kinds of semantic indexing schemes in general; those which completely depend on the external domain knowledge bases for identifying the concepts of terms in the document and those which have some association with some external

knowledge base. Knowledge base is a repository which contains facts in a universe or in certain restricted domain. Ontology is a knowledge base that contains instances of concepts in addition to behavior of the concept - property, relationships, and so on. The proposed method is classified into the semantic approach. It relies on external knowledge base, ontology in particular. It uses domain ontology for the purpose of concept tagging both in the documents and queries, and to identify the relationship among concepts so that it can handle indirect queries and also to have higher accuracy in identifying the proper concepts to reflect the contents of the documents.

2.3 Ontologies

Ontology is one kind of knowledge representation techniques which provides mechanisms to represent knowledge in a general or restricted domain. Introductory descriptions on knowledge representation, basic ontology components, ontology design criteria, ontology tools, and ontology languages are presented as follows.

Knowledge Representation (KR)

According to the authors of [25], "knowledge is data that represents the results of a computer-simulated cognitive process, such as perception, learning, association, and reasoning, or the transcripts of some knowledge acquired by human beings". KR can be defined in terms of the roles it plays; it is a scheme in which substitutes are placed for entities in the world by reasoning the knowledge of the world, and it is also the language in which humans express their understanding about the world [26].

KR is a vital point in AI, natural language understanding, machine learning, and expert systems. In AI, KR is used to formalize and convert the world knowledge into a format which is suitable for machine processing so that other components or agents of the AI system can use it. If knowledge is represented well in AI, then the knowledge will be put into use to derive information that is implied by the knowledge, to converse with people in natural languages, to decide what should come next, to plan future activities, and to solve problems in areas that normally require human expertise. Some of the known KR techniques are semantic network,

rules, and logic [27]. For those indexing schemes which rely on or use external domain knowledge, quality knowledge representation technique must be used to produce high performance indexers.

Ontology to represent knowledge

Different researchers have given different definitions for the term ontology and the definition differs according to the context. The definition of ontology is given from two main perspectives - Philosophy and Artificial Intelligence. From the Philosophy perspective, Ontology is defined as "part of the metaphysics that deals with the being in general and with its transcendental properties" [28]. From the same perspective, philosophers have different senses for the term Ontology and ontology [29]. Ontology is uncountable having no plural form, is addressed by Aristotle, and refers to a discipline that studies the nature of being or theory of existence. It is intended to give answers for questions like *"What is being?"* and *"What characteristics do all beings have in common?*" [30] whereas ontology has a plural form and refers to a system of certain categories responsible for designing some view of the world and the system is independent of the language even if it is reliant on a particular philosophical view [29].

On the other hand, in Artificial Intelligence discipline, many definitions have been given to the term ontology and it has brought controversy to discuss on this issue. Nevertheless, one of the most known ontology definitions is; it is a formal and explicit specification of a conceptualization in a certain domain [29]. The subject of ontology is the study of the categories of things that exist or may exist in some domain. Studying the categories of things imply understanding and reasoning activities of the ontology using a particular language and a specified set of vocabularies. In AI, it is used to denote a particular object rather than a discipline. In this study, the ontology used in intelligent computer applications (i.e., in AI fields) is applied. Therefore, throughout this document ontology refers to a systematic formalization of concepts, definitions, relationships and rules that capture the semantics content of a domain.

Basic components of ontology

Even though the world can be modeled differently having different structure inherited from the used modeling language, the basic components are believed to be the same. The components may be named differently from one ontology to another. Despite this, their core components are largely shared between different ontologies. These components can be categorized in to two groups; those which describe the entities of that particular domain like concepts, instances, and relationships and those that are used to describe the ontology itself. The core components are listed as follows:

- ✓ **Concepts** also called Classes, Types or Universals: are the core components of most ontologies. Concept is a unit of knowledge which is represented by a descriptive statement or a formal expression and its meaning is shared among identified group of responsible persons for the concept's domain [31]. Concept is a collection or types of objects labeled with terms. One concept can be a sub-concept of (also known as sub-class of/kind-of/part-of) another concept. This structure of concepts is usually referred to as concept taxonomies. If a concept c_1 is a sub-concept of a concept c_2, then the individuals of c_1 are also the individuals of c_2. As there are relationships among entities in the real world, there are relationships among concepts of a particular domain as well.
- ✓ **Instances** also called individuals or particulars: are concrete examples of concepts in a domain of interest. Instances represent specific elements attached to a specific concept in the domain ontology. Instances are the `things' represented by a concept. Instances relate with other instances by the relationships shared by the concepts the instances belong to.
- ✓ **Relation**: refers to interactions between concepts or concepts' properties. Relations in an ontology describe the way in which individuals relate to each other. Relation can be defined directly between instances or between concepts. Relations defined between concepts describe the relationship between all instances of these concepts.
- ✓ **Axioms** are explicit rules defined to constrain the use of concepts and the values for classes or instances. They are used to model sentences that are always true.

Types and roles of ontologies

According to [29], on the basis of the level of generality ontologies can be classified into Upper level, Domain, Task, and Application ontologies. Upper level ontologies are developed to describe generic concepts (such as matter, object, event, and time) and to be applicable for any problem without being dependent on any particular domain or problem. Domain ontologies are those designed to represent knowledge in a specific domain. Domain ontologies are developed by specializing concepts generated in upper level ontologies. Task ontologies are modeled to describe concepts for a generic task. Application ontologies are designed to reflect concepts from a particular domain and task.

Basic Criteria for ontology design

As the perception of the people about a certain domain of knowledge is different from one another, the ontologies developed by different persons may not be the same even if the domain is similar. It is not possible to take one modeling technique as a correct technique to model a certain domain because there are lots of viable alternatives [32]. The best model depends on the problem in mind. In fact, it is advisable to consider some valuable criteria that will lead to a better design even if there are different ways to model ontologies.

The following are some of the main design criteria that ontology developers need to consider while designing any kind of ontology [33,34]:

- ✓ **Clarity**: Ambiguity in definition of terms must not exist or at least be minimized while constructing concepts. The motivation to define concepts may come from the community but it must not be influenced by the social situation or the computational context [33]. If a term is defined without satisfying the sufficient information condition then some examples should be included to help readers or users of the ontology understand the concepts or terms specified in it.
- ✓ **Coherence**: The definition of concepts stated using axioms, the natural language documentation, and the examples, all these must be consistent. For instance, a concept definition described using axiom must be consistent with a definition written in Natural language for the same concept or an example written in an informal way to clarify the same concept so that the ontology will not have inconsistency problem.

- ✓ **Extendibility**: As it is stated in [35], ontology should be designed in a way that it can offer a base for a range of anticipated tasks and it should be possible to extend the model. Extending can be defined as adding new terms into the model and giving definition to it based on the vocabulary of the model.
- ✓ **Minimal encoding bias**: there are two levels of representation; knowledge level and symbol level. It is not recommended to use the symbol level representation because different knowledge-sharing agents may use variety of notations or symbols. So knowledge level encoding should be used to model the ontology because it is a common and sharable representation.
- ✓ **Minimum ontological commitment**: An agent is said to be committed to an ontology if its observable actions are consistent with the definitions in the ontology. Ontologies should represent the world being modeled with few states as possible so that knowledge can be shared with and among agents committed to the ontologies easily.

Ontology development methodologies

Since ontology development is not an easy task, people need a sophisticated method to help them build ontologies. Varieties of methodologies have been proposed for ontology building though all are not matured enough. Some of the known approaches are CYC, Uschold and King's method, Gruuninger and Fox, The KACTUS approach, The SENSUS approach, METHONTOLOGY, On-To-Knowledge methodology, and CO4 [36]. The **Uschold and King's** is application independent and straightforward method for ontology development, as it is presented in [13]. Therefore, this method is selected to build the football domain ontology for this thesis work.

Ontology language

Ontology languages define notations to represent ontologies. There are requirements for ontology languages which help to identify the more appropriate language for a certain task. These requirements are a well-defined syntax, a formal semantics, efficient automated support for answering queries, and sufficient expressive power to model the domain of interest. Ontology languages are generally classified into three categories; logical languages, frame based languages, and graph based languages. Logical languages are built based on first order predicate logic, rule based logic, or description logic. Frame based languages are much similar to relational

databases. On the other hand, the graph based languages are emerged for the purpose of building ontologies that are used by the semantic World Wide Web applications.

Ontology languages are also classified into two according to their purpose; classic ontology languages and web-based ontology languages. KIF (Knowledge Interchange Format), Ontolingua, CKML (Conceptual Knowledge Markup Language), F-Logic (Fuzzy Logic), CycL, and OCML (Operational Conceptual Modeling Language) are some of the classic ontology languages [37]. XML (EXtensible Markup Language), RDF (Resource Description Framework), OIL (Ontology Interchange Language), and OWL (Web Ontology Language) are categorized in the web-based ontology languages class [38]. In this study, the OWL language is chosen to develop the football domain ontology since it is the easiest and latest web-based ontology language.

Ontology tools

There are a variety of editing tools for ontology construction. OntoEdit, WebODE, OilEd, SWOOP, TBC, and Protégé are some of the most known ontology development tools [37]. Among all the ontology tools available for free, protégé is selected for building the football domain ontology. Protégé is chosen because it is free, easy to use, and it supports the OWL language as well, and has all the services needed to construct this particular ontology.

2.4 Amharic Language Writing System

In this section, the literatures written on Amharic language that are relevant for this thesis are briefly discussed. The historical details about the language, the alphabets, the punctuations, the numbers, and the grammar of the language are discussed. In addition to these, the possible challenges that may occur while developing automated system which involves representing the Amharic language in machine are indicated.

History about the language

The Amharic language is the national or official language of Ethiopia. It is spoken by around 27 million people as the first language and 7-15 million people as a second language; it is the second most-spoken Semitic language in the world [12]. The Amharic language belongs to the Semtic language class as it is constructed using the script of the Geez language [39].

The Amharic alphabets

The alphabet of the Amharic language consists of 33 core symbols or Fidel (ፊደል). Each of these core symbols occur in seven different orders; the basic character plus six different symbols or orders formed from the basic character. In total, there are 231 distinct symbols in Amharic language [40]. Unlike English language, in Amharic one symbol or character represents both a consonant and a vowel. The six orders are formed by attaching diacritic markings to the basic symbol to combine consonants with vowels. For instance by attaching the "ु" marker to the basic symbol "ሀ" the second order "ሁ" symbol is generated. The Ethiopic script is a syllabary rather than an alphabet because it doesn't have separate characters for vowels unlike alphabets in other language but for the sake of convenience it is named as an alphabet.

Punctuation

In addition to alphabets, Amharic has its own punctuations [41]. There are varieties of punctuation marks used in handwriting but not all of them are mostly applicable in Amharic software. The most known punctuations are:

- ✓ The word delimiter: a colon (:), referred to as "hulet netib" in Amharic, is used to separate words. This punctuation is rarely used in Amharic software.
- ✓ The sentence delimiters: the symbol "four dot" (::) is used as a sentence delimiter which is equivalent to the symbol "." in English.
- ✓ The "Netela serez"symbol ("): its purpose is equivalent to the comma symbol in English.
- ✓ The dirib serez (፡): has same purpose as semi-colon has in English language.

In addition to those listed above, the language uses other punctuation marks taken from other languages such as ? , ! , " , ", ', /, \, and so on.

Numbers

In Amharic writing system, two types of numbering systems are used. The first one is taken from the Geez language. It is believed that the geez language has constructed these numbering system based on the Greek letters [39]. But these Amharic numbers, [42], are not suitable for mathematical computation since there is no symbol for the number zero in this numbering system [41]. It has mostly been used for page numbering and for dates. The second numbering system is the one used in the English language. This one is more suitable to use for automatic Amharic document processing applications.

The Amharic Grammar

Amharic is written from left to right with its own writing system, the typical clause order noun + object + verb [40,43,44]. This order is different from the noun + verb+ object order of the English language. Amharic has inflectional morphological structures, which requires a complex morpheme analyzer for morpheme generation and word formation as well. In Amharic language, one word can be a sentence by implicitly combining the object, subject and verb together. For example, the word መታት can be taken as a sentence; the subject is "እሱ", the object is "እሷ" and the verb is "መታ". Therefore, identifying morphemes from a word is very difficult. Moreover, the plural nouns, plural verb formation in Amharic language is more or less completely different from that of English language.

Challenges in Amharic Language

- ✓ **Different symbols with same pronunciation**. For example, the ሐ, ሀ, and ኀ are pronounced similarly with one another and also with fourth order characters ሐ ሃ and ኃ. As it is stated in [10], historically these characters have different sounds. In automatic Amharic document processing, there should be a mechanism to handle this characteristic of the language.
- ✓ **Different ways of writing the same word:** it is very common to see different people writing the same word in a different way. This spelling variation happens mostly when a word from other language is used without translation which is referred as transliteration. For example, ፕሮምየር ሊግ and ፕሬሚየር ሊግ. These two compound nouns are transliterated

from the English word premier league. When developing any Amharic text processing system, these spelling variations should be handled.

- ✓ **Formation of compound words:** compound words are formed by joining two words with or without modification. For example, the word "ኳስ ሜዳ" is formed from the words "ኳስ" and "ሜዳ" without any change on the words. In another case, for instance, a compound noun ቤተክርስቲያን is formed from two words, ቤት and ክርስቲያን with a little modification on the word ቤት. Moreover, those compound nouns that are formed without modifications can be written with or without a gap between the words like "ኳስ ሜዳ" and "ኳስሜዳ". In spite of this writing variation, with or without a gap, the meaning of the words is the same. Hence, every Amharic application should consider this writing variation in order to capture words with similar meanings.
- ✓ **Changing geez to roman form**: There is no common standard to change the Amharic words written in Geez script into roman font. For instance, "መምታት" can be written as "memitat" or "memetate". It is different from one writer to another. If someone wants to represent Amharic words in roman font for some computer application, he/she has to define his/her own transliteration for each Amharic character as there is no any predefined standard to do so.

2.5 Information Extraction Methods

Since extracting structured information from the news text is one part of this research, some of the most known methods used for Information Extraction (IE) are discussed in this Section. IE is a document or information processing task similar to information retrieval. The major difference between IE and IR is their objective. IE is a query-driven process intended to capture sub-information or factual information from a document based on the user interest whereas IR is a document driven process used to retrieve information contained within a document [45]. There are two main categories of information extraction approaches; Knowledge Engineering Approach and the Automatic Training Approach [46].

- ✓ **Knowledge Engineering Approach**: In the knowledge engineering approach, also known as the rule-based approach, information embedded in a text is extracted using rule

provided by knowledge engineers. The knowledge engineers are supposed to be either experts on the domain or very familiar with the language. Information extraction systems developed using this approach have a very good performance when compared to the automatic training approach because the rules are written manually by experts. The problem of this method is it is labor intensive and time consuming.

- ✓ **Automatic Training Approach**: The automatic training approach, also known as the machine learning approach, learns the rules automatically using an annotated corpus. This approach doesn't need the involvement of knowledge engineers to assist in constructing rules for the extraction purpose. Even though this approach is easy and quick, its performance is much lesser when compared to the rule-based approach.

Chapter Three - Related Work

In this Section, related researches conducted both for Amharic and non-Amharic texts are discussed. First, the reviews on the existing research works on the area of Amharic text Indexing will be presented. Then, the automatic indexing methods proposed for non-Amharic languages are presented.

3.1 Existing Indexing Methods for Amharic Language

Different researchers have tried to build indexing mechanisms for Amharic text using some of the approaches discussed in Section 2.2. Each of these research works has their own strong and weak parts. In this section, some of those methods that have been proposed so far for Amharic text indexing are presented.

Enhanced Design of Amharic Search Engine

There was another attempt made to develop an Amharic search engine, "Design and Implementation of Amharic Search Engine" by Tessema [2]. But the researchers of the "Enhanced Design of Amharic Search Engine" argue that Tessema's work has considerable weaknesses that degraded the performance of the search engine [9]. The reason for conducting this research, Enhanced Design of Amharic Search Engine, is to provide a better search engine for Amharic language by improving Tessema's work.

The indexing part of the system is done using lucene indexer. Lucene uses IDF (Inverse document frequency) approach for indexing. As IDF is a key word based method, the words (i.e., those words which are used to represent documents) may not be capable of reflecting the semantics/contents of the documents. This indicates that there is a need for a semantic indexer for Amharic search engines.

Amharic Text Retrieval

This research is conducted to explore the advantage of latent semantic indexing technique to build semantic indexer for Amharic text documents [3]. The experiment is made to see what improvement the LSI method can bring over the classical exact term matching technique. The main tasks of the experiment consist of preprocessing and indexing, K-dimensional SVD, and Query Projection, Matching and Ranking.

LSI involves computing term weights using log-entropy weighting scheme to produce weighted term-by-document matrix, W. Then, SVD is applied on W using Matlab's built in function, *svds*. During query processing, a query is taken as a pseudo document and projected into the reduced vector space computed using SVD. The Cosine similarity measure is used to compute the distance between a query and a document. The nearest documents to a query, documents with larger cosine coefficient, are selected as relevant documents.

The LSI method has its own advantage and disadvantages. The most important characteristic of LSI method is that it is a mathematical approach, with no need of knowledge about the meaning of the words in the documents. This makes the method generic and completely language independent.

On the contrary, LSI has considerable limitations which affect the performance of the retrieval system. As it is discussed in Section 1.2, the LSI method has a problem of dealing with indirect queries; if the words in the query do not exist in any of the documents, then it may generate irrelevant documents given with rank and miss the relevant ones or give the relevant ones lower rank. Computing SVD whenever a query is posed or a document is added is a very difficult task. Moreover, the matrices generated as a result of the SVD process need a lot of storage space.

N-gram-based automatic indexing for Amharic text

This research was conducted to explore the advantage of applying N-gram-based automatic indexing method for Amharic text retrieval [4]. N-gram-based automatic indexing is a type of statistical approach based on the principle that similar words will have a high proportion of N-grams in common. The basic phases in this method are word identification and weighting, and bigram and trigram generation and weighting. N-grams are sequences of characters or words

extracted from a text. In this study, words and character based adjacent bi and tri-grams were used as index terms. Inverse Document Frequency (IDF) is used for term (i.e., word, bi-gram and tri-gram) weighting. Besides, these terms along with their weights are added to the term vector space. Similarity between a document and a query is computed using cosine correlation formula.

Like the LSI method, the great advantage of this approach is its language independency and its simplicity to use. As the type of the n-gram used in this research is character based, the system can be tolerant to spelling variations and errors in both the queries and the documents. Furthermore, it provides a straightforward ranking of documents that enables the user to see which documents best match the query.

Even though the researchers have proposed to apply the easy and language independent method for Amharic text indexing, their system has its own limitations or weak parts. The major problem of this method is that it could not handle indirect queries because it does not consider the semantics of the terms in the document and in the query. Additionally, the index terms in N-gram based systems are much larger than the word based systems so that it needs a lot of storage. Stemming, one of preprocessing tasks which is known to be used in word based systems, is not applied in this research maybe because the index terms are n-grams. If they had used stemming, it would have reduced the number of unique terms to be indexed. Besides size reduction, stemming helps to group words that have the same concept so that users will not worry about which format they have to use while writing queries. In addition to stemming, stop word removal is not included which would help remove words that may appear frequently in many documents and contribute no good for retrieving relevant documents.

3.2 Existing Indexing Methods for Other Languages

An Ontology-Based Retrieval System Using Semantic Indexing

This research was conducted to build semantic indexing for information retrieval with the support of soccer domain ontology [5]. The ontology is used to construct the index through concept reasoning. The overall system has four components; automated information extraction module, an ontology populator module, an inference module, and a keyword-based semantic

query interface. The information extraction module is a language dependent module which supports only the English language.

Besides its language dependency, this retrieval system has some major weaknesses. The first drawback of this method is that the document-searching process takes much time because the index and the ontology are two separate entities. The second problem is that the index structure is made considering soccer matches only, i.e., the index is built to capture index terms which represent only those news documents that narrate soccer matches. The other main problem is that even though the index is built using concept reasoning, the inferencing method does not compute similarity between original and inferred concepts. In addition to these all, similar weighting method is used both for concepts and concept instances/individuals. As concepts are different from individuals, they should be given different weight values.

Context based Indexing in Search Engines using Ontology

This research was conducted with an objective of building a context based indexer by utilizing domain ontologies [6]. The researchers used ontologies to assist the process of identifying the context of a term in a document. Repository of web page, Indexer, Preprocessing of document, Thesaurus, Context Repository, Ontology Repository, Context of the document, Index, Searcher, and Search Interface are the components of the system.

Those terms that match either with the title of the document or occur frequently in the document are selected as keywords to represent documents. All possible meanings or contexts of these keywords are extracted from thesaurus and context repositories. The context of the keywords will be matched with that of the documents to find the exact context of a keyword in a particular document. In order to identify the context of a document, the keywords/terms of the document and the multiple contexts are compared with the ontologies from the ontology repositories. Finally, the index is built using the keywords, their context and the document IDs.

When users pose a query, they must explicitly add the context of query as well. Then the search engine matches the query terms and their context with that listed in the index. Even though the documents are represented with terms along with their context, if there is no any match found, then nothing will be retrieved. This is a limitation to the system and in addition to this; it does

not respond to complex/indirect queries because only the direct super class of the keyword is selected as the context of the document.

Ontology based text indexing and querying for the semantic web

This research was carried out with an intention of exploring the advantage of ontologies to improve the performance of the indexing and querying components of the information retrieval systems [7]. The proposed system is composed of the following components; ontology importing and mapping, spidering, indexing, ontological indexing, and the front end. The indexing component is intended to extract index terms from the document collections using keyword-based approach. In the ontological indexing part, mapping of words from documents on to the concepts in the ontology is done based on context information. The context of a word in the document is defined using the stems of the words surrounding it and the context of a concept is defined using its super and sub concepts.

The structure of the index is constructed using the spider, the indexer, the ontological indexer and the ontology components. The result of the indexing process is stored in a relational database, PostgreSQL. The database contains tables to store the results of all those components used for index construction.

This method gives equal credits to all the synonyms, sub concepts and super concepts of a concept while comparing the contexts of words and concepts. This may result in retrieving irrelevant documents or giving less relevant documents higher rank. The other drawback of this system is that its incapability of handling indirect queries because it does not apply ontology reasoning techniques.

Evaluating a Conceptual Indexing Method by Utilizing WordNet

In this concept-based indexing method, the benefit of applying WordNet (a domain independent knowledge repository) for the purpose of concept based text indexing was explored [8]. The whole indexing process has concept detection, concept weighting, expansion and indexing phases. Concept detection from a document is done by extracting adjacent terms as multi-term concepts. Then these multi-term candidate concepts along with single-term concepts are mapped onto WordNet. For instance, when a multi-term concept is mapped onto a Synset, all the

synonyms in the synonym Synsets and only one concept in the Hyprenym Synsets that concept belongs to are extracted and passed as an input for the weighting phase.

Concept weighting is done for concepts both in the WordNet and in the document using IDF (i.e., Inverse Document Frequency) method. Expansion is done for those concepts that have synonym concepts from more than one Synset. Finally, the classical keyword-based indexing method was used to construct an index by using the concepts as index terms. The query processing has the same phases as that of document processing. While retrieving documents for a particular query, the index terms generated from a query are matched with the terms in the index.

The advantages of this concept based indexing method are: - it can be used for any domain since WordNet is a domain independent knowledge base and it tries to capture and link synonym terms from documents through mapping and term expanding. On the contrary, it has also some considerable drawbacks. The first one is its incapability of handling indirect queries because of its inability of capturing hyponymy in documents and queries. The other limitation is the time consumption due to the expansion process at run time. Besides these, the fact that the WordNet and the index are two separate entities stored in different places has a negative impact on the speed of the retrieval system. Furthermore, term weighting is done in a similar fashion for both concepts and concept instances/individuals. However, weighting should have been done differently for concepts and individuals since concepts cover higher scope as compared to individuals.

Chapter Four - Design and implementation

4.1 Overview

In this chapter, the design and implementation of the proposed semantic indexer is presented in detail. As depicted in Figure 4-1, the proposed system architecture is composed of three major components: Ontology Development, Document Indexer, and Query Processor. The Ontology development part is composed of designing, building, evaluating, and populating the ontology. The result of this component, football domain ontology, is used later in the document indexer component to build the semantic index.

The major activities during document indexing process are document preprocessing, concept tagging, information extraction, concept weighting, and ontology population. The actual strategy of this study is building an index by integrating it with the domain ontology so as to minimize query processing time and reduce the storage space needed to store the index and the Knowledge Base.

The query processor component of the system is responsible to test the extent to which the proposal made in this work returns relevant information to the user query. The basic tasks employed during query processing are; query preprocessing, creating individuals from query, query tagging, query caching, document retrieval and ranking. All of these tasks are discussed in detail in the following sections.

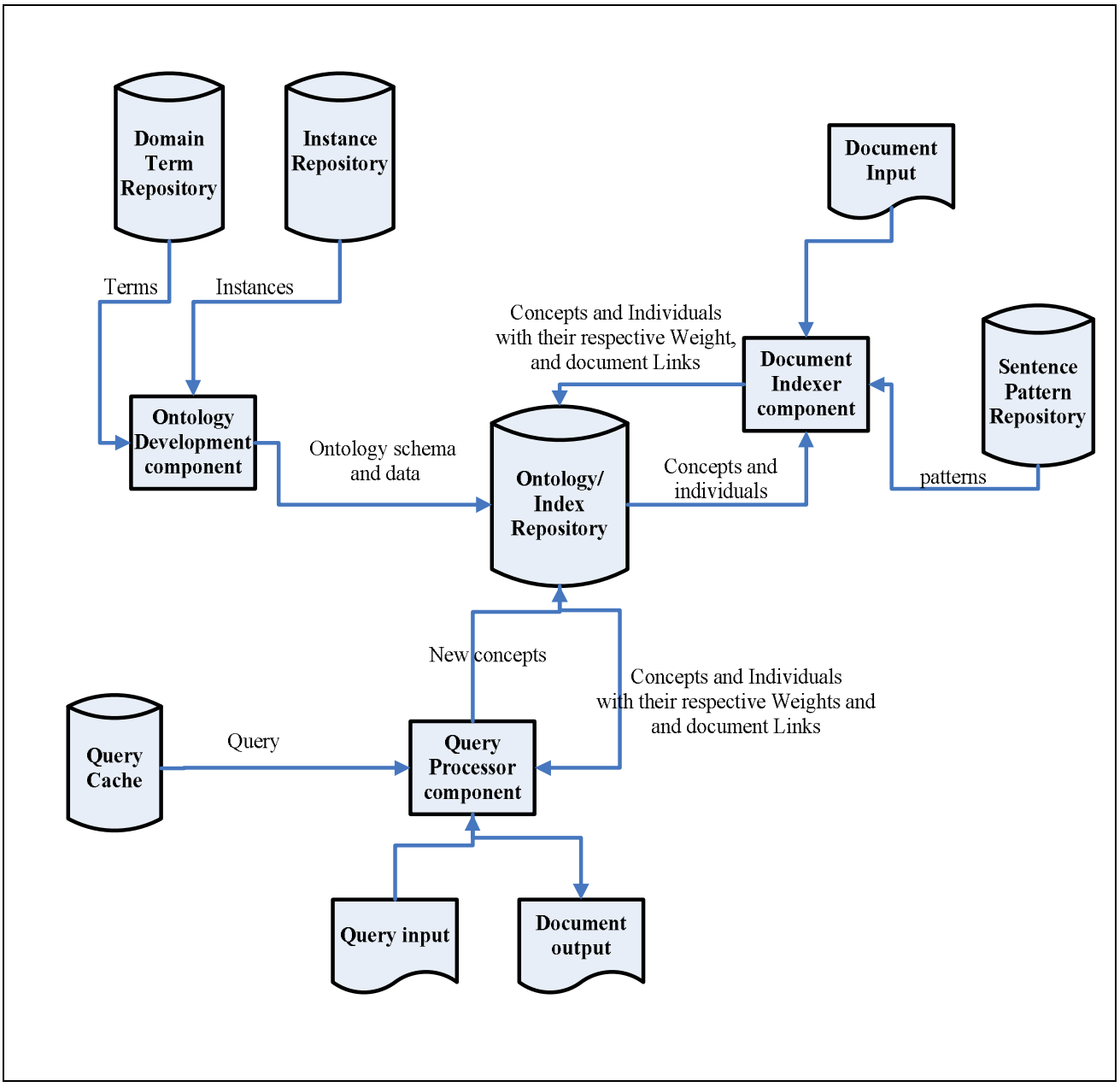


Figure 4-1: General framework of the proposed system

4.2 Ontology Development

Having well structured ontology populated with real data is a precondition for the semantic indexer. Thus, the ontology development is a critical phase in the study and its role is designing and building dedicated football domain ontology. The ontology is built using the Uschold and

King's method [36]. A brief discussion on this method is presented in Section 2.3. The ontology development task is composed of five major processes as described below:

Process 1. Identifying the purpose of the ontology

The ontology is intended to provide a consensual knowledge model of the football domain that will be used in this research work for semantic document indexing. Regarding the scope of the ontology, the most relevant terms to be included are: persons involved in the football world (players, referees, coaches, managers, supporters, fans, and owners), places (Stadiums and player positions) and events (match, competition, player event, and referee event).

Process 2. Building the ontology

This process focuses on creating the structure of the ontology by organizing concepts and relationships. In general there are two crucial steps involved in building ontology- ontology capturing and coding/implementation.

Step 1. Capturing the ontology

Capturing the ontology involves identifying concepts and relations in the football domain with the help of domain expert. In order to identify concepts in the domain the Middle-Out strategy proposed in [33] has been adopted. In this strategy, basic terms are identified first and then they will be specified and generalized as possible. There are 3 activities engaged in this step.

Activity1. **Constructing term glossary**

Glossaries of terms are built to provide the list of domain terms with their full descriptions and synonyms. In this research possible football terms and descriptions are identified and further validated by domain experts[6]. These glossary terms are attached as Annex A.

Activity 2. **Building concept taxonomies**: concept taxonomy is a hierarchical organized collection of concepts. Concepts in the concept taxonomy are related with Hyponym/Hypernym and Meronym/Holonym relationships. Hyponym/Hypernym is also referred to as "is-a" relation -

[6] Domain experts are –experts in Health and Physical Education (HPE) from "Ministry of Education", professionals and managers in information and communication technology from "Ethiopian Football Federation", experts in Sport, Amharic and English subjects from "Wondirad preparatory school", and the secretary of Sport festivals and competitions department in "Ethiopian Football Federation".

a generic relation used to show the relation of class to subclass whereas Meronym/Holonym is also referred to as “is-part-of” relation - used to show the relation of a whole to part. The concept taxonomy built for this particular ontology is shown in Figure 4-2.

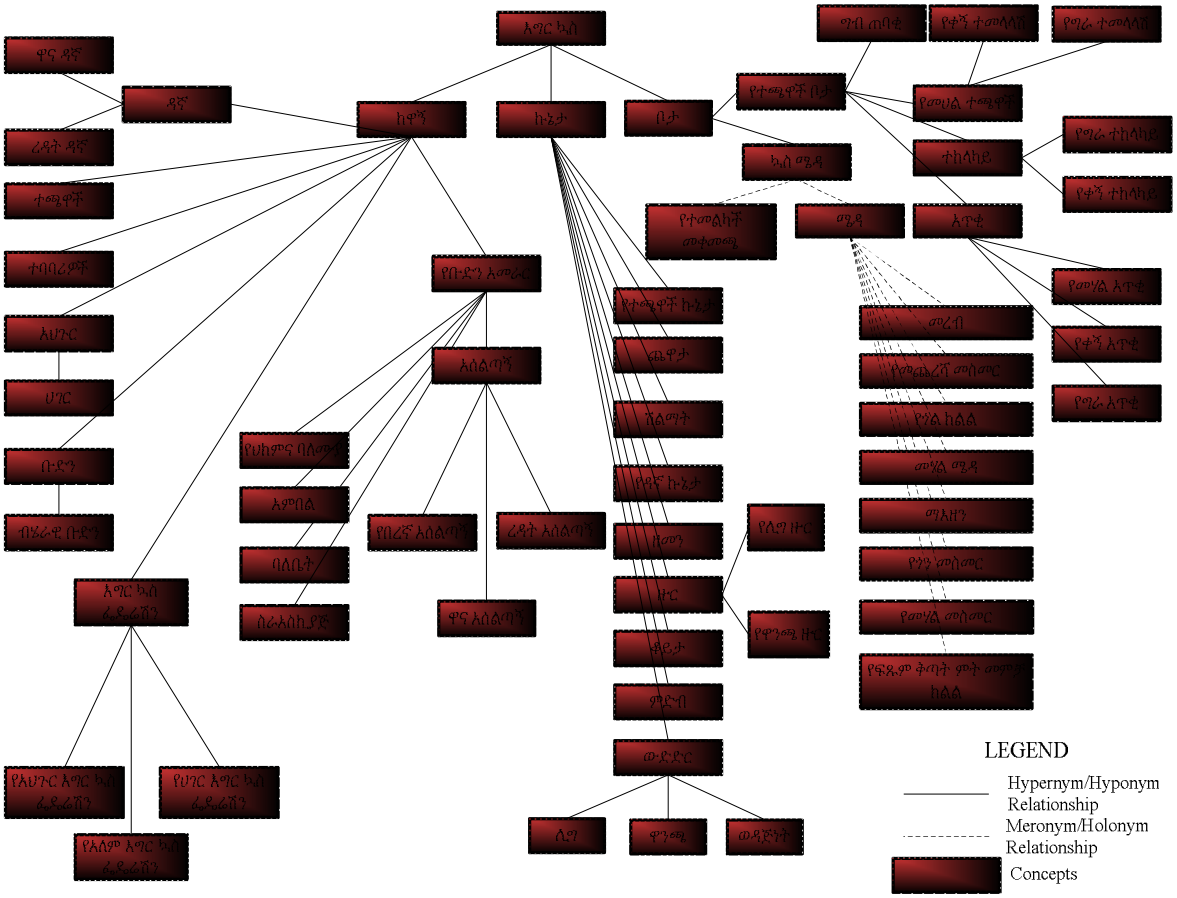


Figure 4-2: The initial concept taxonomy

Activity 3. Identifying concepts relationships: once the concept taxonomy is designed, the possible relationships between concepts are defined. New relationships are identified here in addition to the sub-concept and super-concept relations used to construct the concept taxonomy. These new relationships are used to create more links between concepts. In this research work, we used Web Ontology Language (OWL) based ontology representation. Two OWL relationships are used in this study – ObjectProperties and DatatypeProperties. ObjectProperties

are used to relate objects to other objects whereas DatatypeProperties are applied to create relations between objects and data types. The complete list of Concepts, ObjectProperties and DatatypeProperties used in this research are attached as Annex B, Annex C, and Annex D respectively.

Step2. **Implementation**:

Once the concepts and the relationships are generated manually, the ontology is created using protégé tool and then mapped onto MySQL database. Since it is not possible to write concept names in Amharic while creating ontologies, the concept names and all properties are written in Roman form. For example, the concept "እግር ኳስ" is written as "'gr_kWas". This conversion is done automatically using the converter implemented along with the HornMorpho (Amharic, Oromo and Tigrigna word analyzer) system developed by Michael Gasser [47]. This converter takes a word written in geez alphabets and changes it to Roman form and vice versa.

The concept taxonomy created using protégé, after the concept names are changed to roman form, is shown in Figure 4-3.

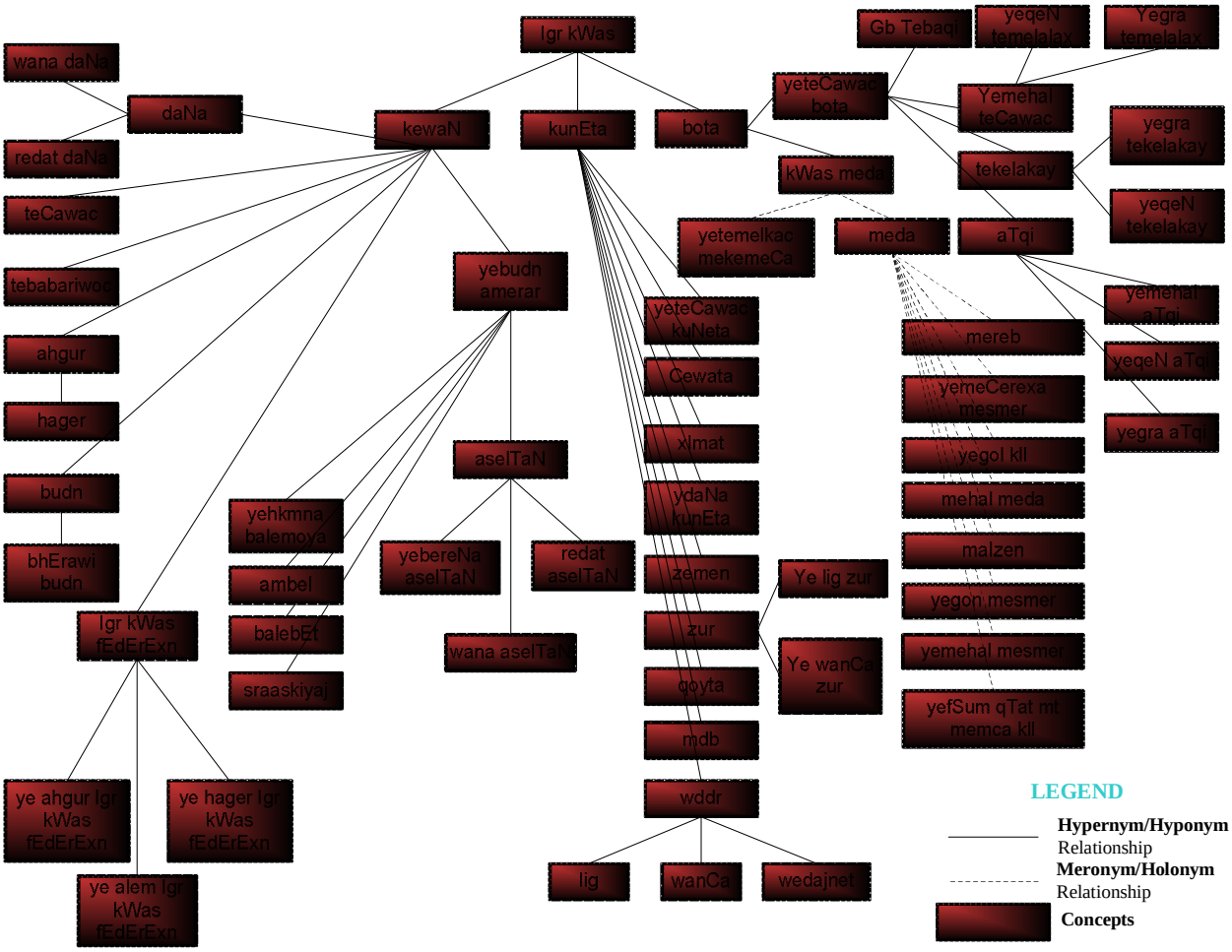
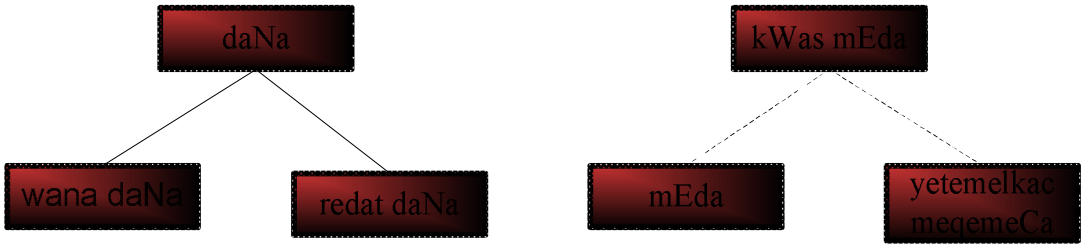


Figure 4-3: The concept taxonomy after the concepts are changed from English to roman form

For example, the concept “ዳኛ”, “ዋና ዳኛ”, “ረዳት ዳኛ” “ኳስ ሜዳ” “ሜዳ” “መቀመጫ” are converted to roman form as “daNa” “wana daNa” “redat daNa” “kWas mEda” “mEda” and “yetemelkac meqemeCa” respectievely. Relationships among these concepts are created as shown below.



The concepts “wana daNa” and “redat daNa” are connected to the concept “daNa” using Hypernym/hyponym relation, i.e., both “wana daNa” (head referee) and “redat daNa” (assistant referee) are kind of “daNa”(referee). Whereas, the concepts “mEda” (Field) and “yetemelkac meqemeCa” (Seat) are part of the concept “kWas mEda” (Stadium).

Process 3. **Evaluating the ontology**

Ontology evaluation is the process of validating the developed ontology by assessing its capability to represent the domain knowledge and be applicable for the semantic indexer developing process. In this research the capability of the ontology in representing facts correctly is assessed by domain experts. These domain experts evaluate all component of the ontology (like its Concept lists, Concept Taxonomy, Datatype Properties, and Object Properties). Based on the response non relevant components are removed and proper adjustments have been made.

Process 4. – Populating the ontology - Adding individuals

Adding instances is the process of inserting individuals and data type values into the ontology. An Individual adder module is developed as part of the ontology development component in order to insert the instances and data type values automatically into the ontology. The information on real concepts instances/individuals, the concept instance data set, were taken from Ethiopian Football Federation (EFF) in order to populate the ontology so that it can be used as a knowledge base. The data is divided in to two sets;

- The first set contains list of individuals and their datatype values for each concept in the ontology. For example, for the concept “ተጫዋች”, the individuals “አዳነ ግርማ”, “ሳልሃዲን ሰይድ”, etc, and the datataype property and value pairs like “ዜግነት” - “ኢትዮጵያዊ”.
- The second data set has the information on relationships between individuals. For example, “አዳነ ግርማ” – “ይጫወታል ለ” –“ኢትዮጵያ ብሄራዊ ቡድን”

Algorithm 4-1 is used to populate the ontology with individuals and Algorithm 4-2 is applied to create relationship among these individuals.

Algorithm 4-1: Individual Adder Algorithm

Line	Individual Adder Algorithm:
1.	**Input**:
2.	Documents: List of individuals with datatype property values // individuals are categorized according to their Type/Class/The concept they belong to. I.e. one document for one category. E.g. Doc1 contains individuals of the concept "ቡድን"
3.	O: Ontology
4.	**Variables**:
5.	RomanizedIndividual: String
6.	IndividualTypePairs: Dictionary
7.	IndividualTtype : String
8.	RomanizedindividualType *:* String
9.	RomanizedDatatype: String
10.	Romanizedvalue: String
11.	**Output**:
	BEGIN
12.	FOR a document D IN Documents
13.	READ Individuals'type/category *//e.g. "ቡድን"*
14.	RomanizedIndividualType= ROMANIZE (Individuals'type) //convert concepts from Geez form to Roman. E.g. "ቡድን" to "budn"
15.	READ Individuals FROM D
16.	FOR each Individual I IN Individuals
17.	RomanizedIndividual = ROMANIZE(I) *//convert individuals from Geez form to roman. e.g. "መከላከያ" to "mekelakeya"*
18.	O.ADD(RomanizedIndividual, RomanizedIndividualType) //create an individual RomanizedIndividual for the concept RomanizedIndividualType. E.g. create individual "መከላከያ" for the concept "ቡድን".
19.	READ datatype-value pairs FOR individual I FROM D
20.	FOR each datatype-value pair IN datatype-value pairs
21.	READ datatype, value FROM datatype-value
22.	RomanizedIndividual = ROMANIZE(I)

23.	RomanizedDatatype = ROMANIZE(datatype)
24.	Romanizedvalue = ROMANIZE(value)
25.	O.ADD(RomanizedIndividual, RomanizedDatatype, Romanizedvalue) //ADD the value *value* for the dataType property *datatype* for the Individual *I*. E.g. for the Indidvidual “የኢትዮጵያ ብሔራዊ ቡድን ” add “ዋልያ” as a value for its datatype property “ቅፅል ስም”
26.	Next
27.	Next
28.	Next
29.	RETURN O
	END

Algorithm 4-2: Relationship Creator Algorithm

Line	**Relationship Creator Algorithm:**
1.	**Input**:
2.	Documents: List of documents which provides information on relationship between individuals.
3.	O: Ontology
4.	**Variables**:
5.	RomanizedIndividual1: String
6.	RomanizedIndividual2: String
7.	RomanizedRelationship: String
8.	ObjPrpty: ObjectProperty
9.	**Output**:
	BEGIN
10	FOR a document D IN Documents
11	FOR each relationshipBetweenIndividuals IN D
12	READ Individuals1 as I1, Individuals2 as I2, relationShip FROM relationshipBetweenIndividuals
13	READ I1’stype, I2’stype FROM O //e.g. “ቡድን”
14	RomanizedIndividual1 = ROMANIZE(I1) *//convert individuals from Geez form to roman. e.g. “መከላከያ” to “mekelakeya”*

15	RomanizedIndividual2 = ROMANIZE(I2) *//convert individuals from Geez form to roman. e.g. "መከለከያ" to "mekelakeya"*
16	RomanizedRelationship = ROMANIZE(relationShip) *//convert individuals from Geez form to roman. e.g. "መከለከያ" to "mekelakeya"*
17	ObjPrpty = GETOBJECTPROPERTY(relationship, O) //get the objectPeroperty with the name *relationship* FROM ontology
18	O.ADD(RomanizedIndividual1, ObjPrpty, RomanizedIndividual2) //create a statement and add it to ontology. E.g. O.ADD("salhadin seyd","yCawetal le", "bherawi budn")
19	Next
20	Next
21	RETURN O
22	**END**

Process 5. Documenting

In this process, every single work done in each phase while building the ontology and the actual product, the ontology, are documented in a clear fashion. Documentation is not a onetime process, i.e., it is done whenever a change is made and documentation is used as a reference during maintenance.

Finally, the ontology is stored in MySQL database. And inferencing will be done by restoring the ontology into memory. Putting the ontology in a database has an advantage of storing huge ontology data.

4.3 Document Indexer

Document indexing is the process of designing the structure of the proposed index and implementing it for the purpose of improving query processing performance. In the context of this research, indexing involves set of activities. The first activity is applying document preprocessing methods on the document collections used as input. Set of concepts existing in each document are identified using the Concept Tagger module. After identifying the concepts,

the structured information from documents is extracted using a knowledge engineering approach (also called rule based information extraction approach). This component exploits the football domain ontology built in the ontology development component to accomplish most of it activities.

The indexer component utilizes the Document Preprocessing, Concept Tagger (CT), Information Extraction (IE), Concept Weighting (CW), and Ontology Population (OP) modules to achieve its purpose. The interaction among these modules is illustrated in Figure 4-4 and discussed in detail as follows.

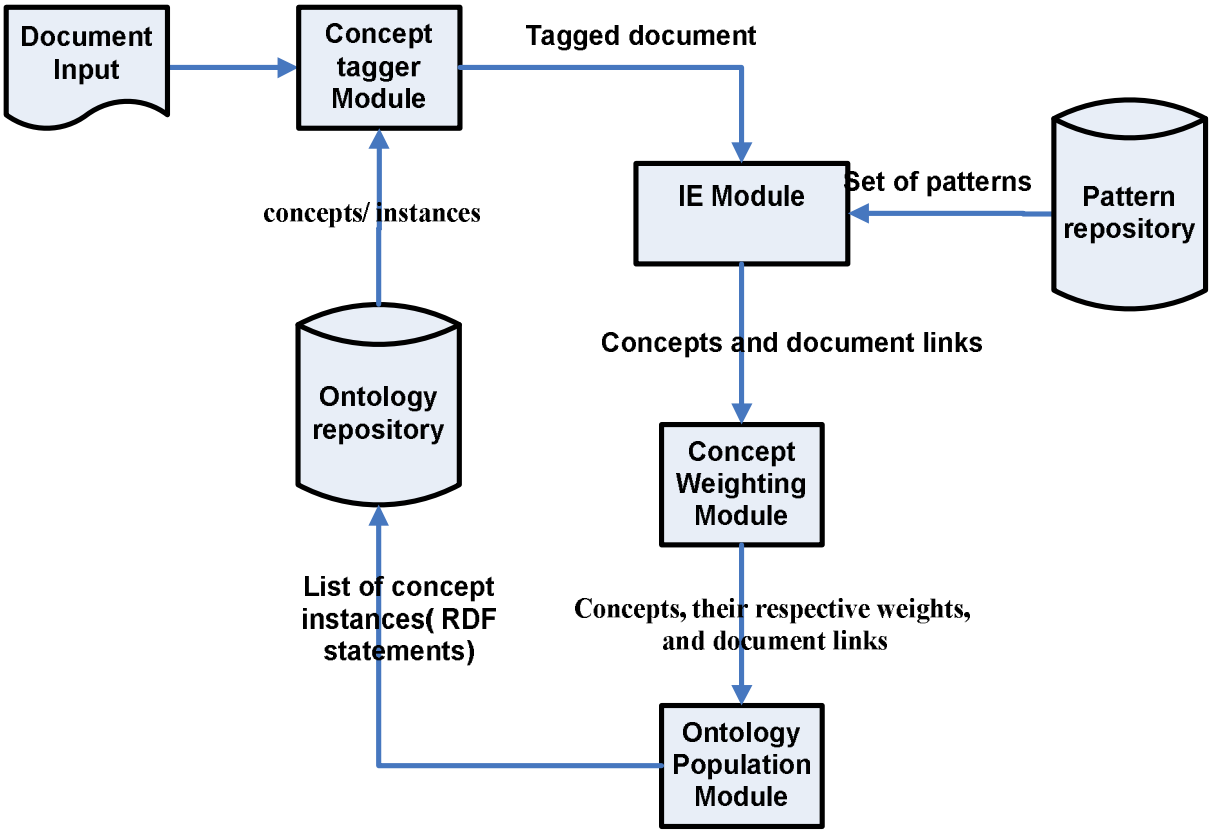


Figure 4-4: The interaction among the modules incorporated in document indexing

Document preprocessing Module

Document preprocessing is the process of making a document ready for indexing by removing unnecessary words and changing characters and words into their common form. As discussed in Section 2.1.4, Amharic language has the following challenges:

1. Different ways of writing the same word (different symbols to write same character)

   In Amharic, there are some characters represented with different symbols and yet have the same sound. In [48], the authors stated that the use of these characters in writing system do not change the meaning of words and thus can be used interchangeably. For example, the symbols ሀ, ሃ, ሐ, ሓ, ኀ, and ኃ have same sound thus all these symbols, whenever they appear in a text, are normalized to ሀ. In addition, all orders of these symbols (the 6 orders) are normalized to corresponding order of ሀ's. Similarly, ሠ and ሰ, አ and ዐ, ፀ and ጸ are groups of symbols with same sound. Each of the symbols in each group shall be converted to one form and their orders as well using the preprocessing module.

2. Formation of compound nouns:

   In [41], the authors showed the difficulty in identifying compound nouns as such nouns can be written either as two separate words or as a single word. For example, the compound nouns "ኳስ ሜዳ" and "ኳስሜዳ" both represent the same entity "Stadium". In order to minimize this difficulty, the popular compound nouns known in the football domain are identified and stored as knowledge to be used appropriately whenever required.

In order to address the challenges presented above we have applied ***normalization*** process. Normalization is the first preprocessing task and those documents on which normalization is applied are called as *"**Normalized documents**"*.

Stemming is the second preprocessing task which is defined as the process of reducing words to their base forms by removing the prefix and suffix forms. In information retrieval systems, stemming provides a means to minimize index terms and hence save storage space and increase the system speed. The copy of the "***Normalized document" set*** is given to the stemmer as input so that ***"normalized and stemmed document"*** set can be produced. The stemmer implemented along with the search engine developed by Tessema [2] was used in this research.

Two sets of documents are produced during document preprocessing; **“Normalized and stemmed document”** and **“normalized but not stemmed document”** sets. Both these sets of documents are later tagged/annotated by the concept tagging (CT module) separately – the methods used for tagging are discussed in the concept tagging module. As a result, another two sets of documents – **“Stemmed and tagged”** and **“Tagged without being stemmed”** are produced and used in the document indexing process.

The first set, **“Stemmed and tagged**”, is used by the concept weighting (CW) module to capture all concepts and individuals that occur in the document collections. This set of documents is chosen for concept identification because both concepts and individuals can be written in different form so that they should be stemmed. For example the concept “ዳኛ” can be written as “ዳኞቻችን” in a document. All the words in the document collection must be stemmed because those written with suffix or prefix should not be missed.

The second set of documents, **“Tagged but not stemmed”,** is used later by the Information Extraction (IE) module to extract embedded information -- the format of the rule used for information extraction purpose is discussed in detail in IE module. This document set is chosen instead of the first set, **“Stemmed and tagged**”, due to the reason that applying pattern matching on stemmed document is difficult. This happens because in some cases, it is not possible to extract the information embedded in a sentence. For example, in the sentence - “መከላከያ በደደቢት 2ለ1 ተሸነፈ”, if the words in the sentence are stemmed, the character በ will be removed from the word በደደቢት and then identifying the winner and loser teams will be impossible. Therefore, **“Tagged but not stemmed**”, documents are used instead of the **“Tagged and stemmed”** document collections, for information extraction process.

Concept Tagger (CT) module

The concept tagger module is responsible for annotating documents, both in the **“Tagged but not stemmed**” and **“Tagged and stemmed”** document sets, semantically using instances (i.e., individuals) from ontology. CT module annotates document using two methods: by applying rules and by using concepts stored in the ontology. These two methods are discussed below.

a. Using rules to tag concepts

Rules are defined using patterns which are combinations of numbers and literals used to represent concepts. Patterns are built in order to identify individuals of concepts from a text. Regular expressions are used to construct these patterns. The main components of regular expressions used to build these patterns are:

- Numbers. E.g., the numbers 0 to 9 represented as [0-9] in the pattern [0-9]+ ኛ (ሳምንት|ዙር)
- Literals. E.g. “ነጥብ” in the pattern “[0-9] + ( )*ነጥብ”.
- The Symbol “*”. It is used when it is necessary to set characters as optional. I.e. if there are characters that can happen zero or many times.
- The Symbol “|”. It has equivalent meaning with the word “OR”

The rules have patterns particularly for concepts: Round (“ዙር”) Point (“ነጥብ”), Result (“ዉጤት”), and Rank (“ደረጃ”). For example, instances/individuals of the concept Round (such as “15ኛ ሳምንት”, “14ኛ ዙር”, etc,) are captured using the pattern “[0-9]+ኛ (ሳምንት|ዙር)”.

Table 4-1 shows rules generated with the help of domain experts and specified with regular expressions built to represent facts. All the other concepts in documents are tagged using the concepts from the ontology.

Table 4-1: Patterns used to tag concepts in a document

Concepts	Patterns
“ዙር”	[0-9]+ኛ (ሳምንት|ዙር)
“ነጥብ”	“[0-9]+( )*ነል”
“ዉጤት”	'[0-9]+( )*ለ( )*[0-9]+'
“ደረጃ”	[0-9]+( )*ነጥብ

b. Using concepts from ontology to tag concepts

Since it is impossible to generate patterns for most of the concepts in the domain, it is necessary to refer to the individuals of the concepts populated in the ontology. For example, it is a fact that we cannot find a pattern which can represent all of the individuals of the concept "ቡድን" (i.e.,

“ቅዱስ ጊዮርጊስ” መከላከያ” “ደደቢት”…) because names cannot be represented using regular expressions. Therefore, individuals of such kind of concepts are annotated using concept information stored in the ontology.

Two different algorithms are used to tag the two sets of documents - “Normalized but not stemmed” and “Normalized and stemmed”. When tagging the first set of documents, the individuals from the ontology are used without being stemmed. On the contrary, Stemmed individuals are used to annotate documents from the “Normalized and stemmed” document sets. Besides this, all the words in each of the patterns in the pattern collection are also stemmed.

Algorithm 4-3 is used to annotate documents from “Normalized but not stemmed” document set.

Algorithm 4-3: Tagging Algorithm1

Line	**Tagging Algorithm1:**
1.	**Input**:
2.	Documents: List of **“normalized but not stemmed”** documents
3.	O: Ontology
4.	PC: pattern collection *// PC is the one listed in Table 4-1*
5.	**Variables**:
6.	GeezifiedIndividual: String
7.	IndividualTypePairs: Dictionary
8.	IndividualTtype : String
9.	*GeezifiedndividualType :* String
10.	ConceptPatternPairs : Dictionary
11.	**Output**:
	BEGIN
12.	READ Individuals FROM O
13.	FOR each Individual I IN Individuals
14.	GeezifiedIndividual = GEEZIFY (I) *//convert individuals from Roman form to Geez. e.g. “mekelakeya” to “መከላከያ”*
15.	IndividualTtype = GetIndividualType(I, O) *//type of individuals means the classes/concepts that the individuals belong to. E.g. The type of the individual “mekelakeya” is “budn”*

16.	GeezifiedndividualType= GEEZIFY(IndividualTtype) *//convert concepts from Roman form to Geez. E.g. "budn" to "ቡድን"*
17.	IndividualTypePairs.ADD(GeezifiedIndividual,GeezifiedndividualType) // add "individual" and "type" pair to dictionary IndividualTypePairs. E.g. IndividualTypePairs.ADD("መኪላኪያ","ቡድን")
18.	READ rows FROM PC
19.	FOR each row IN rows
20.	READ Concept, Pattern // read the values in each colons in the current row E.g. Concept = "የ ሊግ ዙር", Pattern = "[0-9]+ኛ (ሳምንት|ዙር)"
21.	ConceptPatternPairs.ADD(Concept, Pattern)
22.	FOR document D IN documents
23.	FOR a sentence S IN D
24.	FOR each Concept-Pattern Pair IN PC
25.	READ Concept, Pattern FROM Concept-Pattern Pair
26.	IF Pattern has a match M IN S THEN
27.	FIND M IN S and REPLACE it with "<" + M + ">" + "[" + C +"]" // example- [17ኛ ሳምንት]<የሊግ ዙር>
28.	Next
29.	FOR each Individual-Type IN IndividualTypePairs
30.	READ Individual, Type FROM Individual-Type
31.	IF Individual is IN S THEN
32.	FIND Individual IN S and REPLACE it with "< + Individual + ">" + "[" + Type +"]" // example- [መከላኪያ]<ቡድን>
33.	Next
34.	Next
35.	Next
36.	RETURN Documents
37.	**END**

Algorithm 4-4 is used to tag documents in the "tagged and stemmed" document set.

Algorithm 4-4: Tagging Algorithm2

Line	**Tagging Algorithm2:**
1.	**Input**:
2.	Documents: List of **"normalized but not stemmed"** documents
3.	O: Ontology
4.	PC: pattern collection *// PC is the one listed in Table 4-1 with the words in the pattern being stemmed so as to capture stemmed words e.g."15ኛ " is stemmed and in a document as "15ኛ"*
5.	**Variables**:
6.	GeezifiedIndividual: String
7.	IndividualTypePairs: Dictionary
8.	IndividualTtype : String
9.	*GeezifiedndividualType :* String
10.	ConceptPatternPairs : Dictionary
11.	**Output**:
	BEGIN
12.	READ Individuals FROM O
13.	FOR each Individual I IN Individuals
14.	GeezifiedIndividual = GEEZIFY (I) *//convert individuals from Roman form to Geez. e.g. "mekelakeya" to "መከላከያ"*
15.	StemmedGeezInd=STEMM(GeezifiedIndividual) //e.g. *STEMM("መከላከያ")* = "መከላከይ"
16.	IndividualTtype = GETINDIVIDUALTYPE(I, O) *//type of individuals means the classes/concepts that the individuals belong to. E.g. The type of the individual "mekelakeya" is "budn"*
17.	GeezifiedndividualType= GEEZIFY(IndividualTtype) *//convert concepts from Roman form to Geez. E.g. "budn" to "ቡድን"*
18.	IndividualTypePairs.ADD(StemmedGeezInd,GeezifiedndividualType) // add "individual" and "type" pair to dictionary IndividualTypePairs. E.g. IndividualTypePairs.ADD("መከላከይ","ቡድን")
19.	READ rows FROM PC
20.	FOR each row IN rows
21.	READ Concept, Pattern // read the values in each colons in the

	current row E.g. Concept = “ዙር”, Pattern = “[0-9]+ኝ (ሳምንት|ዙር)”
22.	ConceptPatternPairs.ADD(Concept, Pattern)
23.	FOR document D IN documents
24.	FOR a sentence S IN D
25.	FOR each Concept-Pattern Pair IN PC
26.	READ Concept, Pattern FROM Concept-Pattern Pair
27.	IF Pattern has a match M IN S THEN
28.	FIND M IN S and REPLACE it with “<” + M + “>” + “[” + C +”]” // example [15ኝ ሳምንት]<የሊግ ዙር>
29.	Next
30.	FOR each Individual-Type IN IndividualTypePairs
31.	READ Individual, Type FROM Individual-Type
32.	IF Individual is IN S THEN
33.	FIND Individual IN S and REPLACE it with “< + Individual + “>” + “[” + Type +”]” // example [መከላከይ]<ቡድን>
34.	Next
35.	Next
36.	Next
37.	RETURN Documents
38.	**END**

The following example shows how Algorithm 4-3 works.

Example:

A document containing the sentence “ሐዋሳ ላይ መከላከያን ያስተናገደው ሲዳማ ቡና 2 ለ1 አሸንፎአል” is used as input to the Tagging Algorithm - Algorithm 4-3. The following individuals are captured and tagged using both methods – rules and concept information.

- The individual “መከላከያ” is identified by referring to the ontology and tagged as [መከላከያ]<ቡድን>
- The individual “ሲዳማ ቡና” is identified by referring to the ontology and tagged as [ሲዳ ማ ቡና]<ቡድን >

- Finally, using the pattern for “ውጤት”, “2 ለ1” is identified and tagged as [2 ለ1]<ውጤት>

As a result, the sentence will be annotated or tagged as ሐዋሳ ላይ [መከላከያ]<ቡድን>ን ያስተናገደው [ሲዳማ ቡና]<ቡድን>[2 ለ1]< ውጤት> አሸንፎአል.

Algorithm 4-4 is applied on the “Normalized and Stemmed” documents after the information extraction process. This is done because of the reason that the newly extracted individuals, by the IE module, should be added to the ontology and used later for concept weighting,

Information Extraction (IE) module

Information extraction module is responsible for extracting information embedded in the document using a rule based approach. Rules are constructed using patterns. Pattern is a combination of concepts and words built to capture the semantics of a sentence. Building pattern is a manual process done with the help of language and sport experts. The documents tagged by the CT module were given to the experts (i.e., language experts and journalists) and they have derived the possible patterns from each sentence in the documents. The following components are used to make up a pattern;

- Concepts – concepts are those tagged ones in the sentence
- Words and Characters
- Symbols like “*”, “.”, “|” – the Symbol “.” Represents any character, “*” denotes number of occurrence which is 0 – Many times, and “|” has equivalent meaning with the word OR.

The steps followed so as to build a pattern from a sentence in a tagged document are shown in the example presented below.

Example:

- Let’s assume the sentence “በ [16ኛው ሳምንት] <የሊግ ዙር> የ [ኢትዮጵያ]<ሀገር> [ፕሪሚየር ሊግ]<ሊግ> ጨዋታዎች ሐዋሳ ላይ [ሐዋሳ ከነማ]<ቡድን>1 [ሙገር ሲሚንቶ]<ቡድን>2ን [1 ለ 0]<ዉጤት> አሸነፈ..” is in a tagged document D.

- Extract all the tagged concepts (<የሊግ ዙር>, <ሀገር>, <ሊግ>, <ቡድን>, <ቡድን>, and <ዉጤት>).
- Extract words and characters (if their meaning affects the semantics of the whole sentence) - words (“አሸነፈ”), and Characters (“ን”).
- Find other words with similar meaning with those extracted in the previous step, (“አሸነፈ”), like “ረታ” and “ቀጣ” and combine these words with these synonym terms using the symbol “|”as (“አሸነፈ|ረታ|ቀጣ”).
- Represent the rest of the words in the sentence using the symbols “.” And “*” together as “.*” which means “any character any number of times” for example in between the concepts <ሊግ> and <ቡድን>1 there are words (ጨዋታዎች ሐዋሳ ላይ) and these words should be represented as “.*”.
- Finally, combine the results of all the previous steps and form the pattern as [<የሊግ ዙር>.* <ሀገር><ሊግ>.*<ቡድን>1 <ቡድን> ን <ዉጤት> “አሸንፎአል/አሸነፈ"]

The possible information which can be extracted from sentences is classified into 5 categories – information on Match-Result, Match-Player-Event, Match-Referee-Event, Competition-Team-Rank, and Competition-Player-Point. Set of patterns were built for each of the categories. The detailed description about these categories is presented below.

- Match-Result**:** in this category patterns that describe the result of a match between two teams are classified. The information extracted using patterns from this category is populated in the following template.

Template 4-1: A template used to populate information in the Match-Result category

ዘመን	ውድድር	ዙር	ከተማ	ኳስ ሜዳ	ሀገር1	ቡድን1	ሀገር2	ቡድን2	ጨዋታ	የቡድን1 ዉጤት	የቡድን2 ዉጤት

The entries “ዘመን”, “ሀገር”, “ዙር”, “ከተማ”, “ኳስ ሜዳ”, “ሀገር1”, and “ሀገር2” are optional. For example, [<ቡድን>+<ቡድን>ን+ <ዉጤት> + "አሸንፎአል/አሸነፈ"] is one of the patterns in this category. In some sentences only one team can be mentioned like the sentence - “የ አዲስ

አበባው ቅዱስ ጊዮርጊስ ቡድን በቅርብ ባደረጋቸው 3 ጨዋታዎች በተመሳሳይ ውጤት 3ለ0 አሸንፏል". In order to capture the information from such kind of sentences, in addition to the other optional entries, we made the entry Team2 also optional. The information extracted from the above sentence looks like: – ጨዋታ = ቡድን1 (ቅዱስ ጊዮርጊስ) vs. ቡድን2 (unknown) with ውጤት = [3ለ0].

- Match-Player-Event: this category refers to patterns that describe events made by a player during a match. The information extracted using patterns from this category is populated in the following template.

Template 4-2: A template used to populate information in the Match-Player-Event category

ዙር	ውድድር	ከተማ	ኳስ ሜዳ	ሀገር1	ቡድን1	ሀገር2	ቡድን2	ቆይታ	ደቂቃ	ተጫዋች	ቡድን	የድርጊት አይነት

Note that Event Type is the different types of actions performed by players during a match such as ጎል (Goal), ኳስ ማዳን (Save), ማሳለፍ (Pass),የማእዘን ምት (Corner kick), ነጻ ምት (Free kick), የጎል ምት (Goal Kick), ኦፍሳይድ(Offside) and so on.

The entries "ዙር", "ውድድር", "ከተማ", "ኳስ ሜዳ", "ሀገር1", "ቡድን1", "ሀገር2', "ቡድን2", "ቆይታ", "ደቂቃ", and "ቡድን" are optional. For example, the pattern "<የ ሊግ ዙር> +< ሀገር> + <ሊግ> + <ከተማ> +<ኳስ ሜዳ> +<ቡድን> +<ቡድን>+ <ቆይታ> + <ደቂቃ> + <ተጫዋች>+ ለ<ቡድን> + <ጎል>"is classified in this category.

- Match-Referee-Event: this category represents patterns of statements that talk about an event made by a referee during a match. The information extracted using patterns from this category is populated in the following template.

Template 4-3: A template used to populate information in the Match-Referee-Event category

ዙር	ውድድር	ከተማ	ኳስ ሜዳ	ሀገር1	ቡድን 1	ሀገር2	ቡድን 2	ቆይታ	ደቂቃ	ዳኛ	ቡድን	ተጫዋች	የድርጊት አይነት

Note that in this category, Event type indicates any of the different types of actions performed by referees during a match such as "ማስጠንቀቂያ መስጠት" (giving warning), "ቀይ ካርድ ማሳየት" (showing red card), "ቢጫ ካርድ ማሳየት" (showing yellow card) and so on.

The entries ዙር, ውድድር, ከተማ, ኳስ ሜዳ, ሀገር1, ቡድን1, ሀገር2, ቡድን2, ቆይታ, ደቂቃ, and ቡድን are optional tags. Besides, the tags ዳኛ and ተጫዋች can be optional but not simultaneously (i.e., either the name of the referee, who performed the action, or the name of the player, whom acted upon, should be mentioned in the sentence). The pattern"<የሊግ ዙር>+ <ሊግ>+<ከተማ>+<ኳስ ሜዳ> +<ሀገር> + <ቡድን> +< ሀገር> + < ቡድን>+<ቆይታ> + <ደቂቃ> + <ዳኛ> +<ቡድን>+ <ተጫዋች+ <የዳኛ ኩኔታt>is one of the patterns in this category.

- Competition-Team-Rank: a collection of patterns of statements that talk about the rank (result) of teams in a particular competition. The information extracted using patterns from this category is populated in the following template.

Template 4-4: A template used to populate information in the Competition-Team-Rank category

ዘመን	ዙር	ሀገር	ውድድር	ቡድን	ደረጃ

The entries "ዘመን", "ዙር", and "ሀገር" are optional tags. The pattern "<ዘመን> <የሊግ ዙር><ሀገር><ሊግ><ቡድን>+ <ደረጃ>" is a pattern in this category.

- Competition-Player-Point: a collection of patterns of statements that talk about the point (number of goals) strikers scored in a particular competition. The information extracted using patterns from this category is populated in the following template.

Template 4-5: A template used to populate information in the Competition-Player-Point category

ዘመን	ዙር	ሀገር	ውድድር	ቡድን	ተጫዋች	ጎል

The entries ዘመን, ዙር, and ሀገር are optional tags. The pattern <ዘመን> + <የሊግ ዙር> + <ሀገር> + <ሊግ> + "ኮ ከ ብ ግ ብ አ ግ ቢ" + <ቡድን> + <ተጫዋች> + <ነጥብ> is in this category.

All the patterns in each of the categories are attached as Annex E The following algorithm, Algorithm 4-5, is applied for information extraction process.

Algorithm 4-5: Information Extraction Algorithm

Line	**Information Extraction Algorithm:**
1.	**Input**:
2.	Documents: List of **"Tagged"** documents
3.	PC : List of patterns from pattern repository
4.	Templates: list of the 5 templates(Template 4-1, Template 4-2, Template 4-3, Template 4-4 and Template 4-5)
5.	**Variables**:
6.	Match : String
7.	**Output**:
	BEGIN
8.	FOR each Document D IN Documents
9.	FOR a Sentence S IN D *// S is a tagged/annotated sentence in the document D*
10.	FOR a Pattern P IN PC
11.	IF S MATCHES with P THEN
12.	IF P is IN Match-Result category THEN
13.	READ ዘመን, ወድድር, ዙር, ከተማ, ኳስ ሜዳ, ሀገር1, ሀገር2, ቡድን1, ቡድን2, ዉጤት FROM S
14.	Match(ጨዋታ) = ቡድን1 + "vs." + ቡድን2
15.	READ የቡድን1ዉጤት, የቡድን2ዉጤት FROM ዉጤት
16.	ADD ዘመን, ወድድር, ዙር, ከተማ, ኳስ ሜዳ, ሀገር1, ሀገር2, ቡድን1, ቡድን2, Match(ጨዋታ), የቡድን1ዉጤት, የቡድን2ዉጤት TO **Error! Reference source not found.**
17.	IF P is IN Match-Player-Event category THEN
18.	READ ዙር, ውድድር, ከተማ, ኳስ ሜዳ,ሀገር1, ቡድን1, ሀገር2, ቡድን2, ቆይታ, ደቂቃ, ተጫዋች, ቡድን, የድርጊት አይነት
19.	ADD ዙር, ውድድር, ከተማ, ኳስ ሜዳ,ሀገር1, ቡድን1, ሀገር2, ቡድን2, ቆይታ, ደቂቃ, ተጫዋች, ቡድን, የድርጊት አይነት TO Template 4-2
20.	IF P is IN Match-Referee-Event category THEN
21.	READ ዙር, ውድድር, ከተማ, ኳስ ሜዳ,ሀገር1, ቡድን1, ሀገር2, ቡድን2,

	ቆይታ, ደቂቃ, ዳኛ, ቡድን, ተጫዋች, የዳኛ ኩኔታ FROM S
22.	ADD ዙር, ውድድር, ከተማ, ኳስ ሜዳ,ሀገር1, ቡድን1, ሀገር2, ቡድን2, ቆይታ, ደቂቃ, ዳኛ, ቡድን, ተጫዋች, የዳኛ ኩኔታ TO Template 4-3
23.	IF P is IN Competition-Team-Rank category THEN
24.	READ ዘመን, ዙር, ሀገር, ውድድር, ቡድን, ደረጃ FROM S
25.	ADD ዘመን, ዙር, ሀገር, ውድድር, ቡድን, ደረጃ TO Template 4-4
26.	IF P is IN Competition-Player-Point category THEN
27.	READ ዘመን, ዙር, ሀገር, ውድድር, ቡድን, ተጫዋች, ጎል FROM S
28.	ADD ዘመን, ዙር, ሀገር, ውድድር, ቡድን, ተጫዋች, ጎል TO Template 4-5
29.	Next
30.	Next
31.	Next
32.	RETURN Templates
33.	**END**

The following scenario shows how Algorithm 4-5 is applied to extract the information embedded in the sentence of a tagged document.

Sentence S: - በ [16ኛው ሳምንት] <የሊግ ዙር> የ [ኢትዮጵያ]<ሀገር> [ፕሪሚየር ሊግ]<ሊግ> ጨዋታዎች [ሐዋሳ]<ከተማ> ላይ [ሐዋሳ ከነማ]<ቡድን>1 [ሙገር ሲሚንቶ]<ቡድን>2ን [1 ለ 0]<ዉጤት> አሸንፏል. The following activities are performed by the algorithm to extract information from the sentence S.

- Look for patterns in the Pattern Collection, PC, which matches with sentence S.
  - The Pattern, [<የሊግ ዙር>.*<ሀገር><ሊግ>.*<ከተማ>.*<ቡድን>1<ቡድን>ን<ዉጤት> "አሸንፎአል/አሸነፈ"] is found in the PC.
- Check the category this pattern belongs to
  - The category, Match-Result is given as an answer
- Read the tags - የሊግ ዙር, ሀገር, ሊግ, ከተማ, ቡድን1, ቡድን2, and ዉጤት
  - the values for the tags of this particular sentence are – "16ኛ ው ሳ ምን ት", "ኢ ት ዮ ጵ ያ", "ፕ ሪ ሚ ያ ር ሊ ግ", "ሐ ዋ ሳ", "ሐ ዋ ሳ ከ ነ ማ", "ሙ ግ ር ሲ ሚ ን ቶ", and "1 ለ 0" respectively
- Create a match as "ቡድን1 vs. ቡድን2"
  - Match (ጨዋታ)= "ሐዋሳ ከነማ vs. ሙገር ሲሚንቶ"

- Read ቡድን1ነጥብ and ቡድን2ነጥብ
  - ቡድን1ነጥብ =1 , ቡድን2ነጥብ = 0
- Fill Template 4-1 with the values of የሊግ ዙር, ሀገር, ሊግ, ከተማ, ቡድን1, ቡድን2, ጨዋታ, ቡድን1ነጥብ, and ቡድን2ነጥብ.
  - The Template is filled as shown below and the value for the entry Stadium is left blank because it is not available in the sentence and it is optional.

ዘመን	ውድድር	ዙር	ከተማ	ኳስ ሜዳ	ሀገር1	ቡድን1	ሀገር2	ቡድን2	ጨዋታ	የቡድን1 ዉጤት	የቡድን2 ዉጤት
	ፕሪሚየር ሊግ	16ኛው ሳምንት	ሐዋሳ		ኢትዮጵያ	ሐዋሳ ከነማ		መገር ሲሚንቶ	ሐዋሳ ከነማ vs መገር ሲሚንቶ	1	0

Concept Weighting (CW) Module

In this module, weighting will be done for the concepts and individuals in each document in the "**Tagged and Stemmed**" document collections. The reason for choosing this document collection instead of the "**Tagged but not Stemmed**" is discussed in the document preprocessing part. The weights computed in this module are used in the document ranking module of the Query Processor component.

The approach used in this study for concept weighting is developed by modifying the classical TF/IDF method used in lucene indexer [2]. The changes made on the TF/IDF technique are;

- Normally in lucene's index term weighting process, all terms in the document collection, excluding stop words, are selected as index terms so that weighting can be done for them. But in this research, concepts and individuals from documents are selected as index terms using the domain ontology.

  For example, from a document with the sentence "[ኢትዮጵይ]<ሀገር> [ፕሪሚየር ሊግ] <ሊግ> ጨዋት ዛር [ኢትዮጵይ ባንክ] <ቡድን>1 እን [መከላከይ] <ቡድን>1 እንዲሁም [ቅዱስ ጊዮርጊስ] <ቡድን>1 እን

[ሲዳም ቡን] <ቡድን>1 ይጋጠማል", using stemmed collection of concepts and individuals loaded from the ontology, the following items (concepts and individuals) are identified.

Individuals: - ኢትዮጵያ of type ሀገር, ፕሪሚየር ሊግ of type ሊግ, and ኢትዮጵያ ባንክ, መከላከያ and ሲዳማ ቡና of type ቡድን.

Concepts: - ሊግ and ጨዋታ.

- The methods used to compute weights for individuals should be different from that of concepts. This is due to the fact that a concept represents a lot of instances hidden with it whereas an individual represents only itself. For example, the concept "ቡድን" and the individual "መከላከያ" appear in the same document. The concepts "ቡድን" represents all of its individuals indirectly including "መከላከያ" however the individual "መከላከያ" indicates only itself. Therefore, these two terms (i.e., "ቡድን" and "መከላከያ") should be weighted differently. The formula used to compute weight for individuals is shown in Equation 4-1. It is similar with the one used in the lucene indexer.

$$\mathbf{IF\_IDF_{i,d}} = \mathbf{IF_{i,d}} \times \mathbf{\log IDF_i} \qquad \textbf{[4-1]}$$

On the other hand, the formula used in this study to compute weight for concepts is shown in Equation 4-2.

$$\mathbf{CF\_IDF_{c,d}} = \mathbf{CF_{c,d}} \times \mathbf{\log IDF_c} \qquad \textbf{[4-2]}$$

Where:

$\mathbf{CF_{c,d}}$ is the occurrences of a concept c in document *d* and it is computed as shown in Equation 4-3.

$$\mathbf{CF_{c,d}} = \mathbf{Count(c)} + \sum_{i \epsilon c} \boldsymbol{Count(i)} \qquad \textbf{[4-3]}$$

$\mathbf{IDF_c}$ is the inverse document frequency – the number of documents divided by the number of documents in which either the concept c or any one of c's individuals occur.

Algorithm 4-6 is used to compute weight for individuals in a document.

Algorithm 4-6: Individual Weighting Algorithm

Line	Individual Weighting Algorithm:
1.	**Input**:
2.	Documents: List of **"Stemmed and Tagged"** documents
3.	**Variables**:
4.	UniqueIndividualsInD: List
5.	AllIndividualsInD: List
6.	DocFreq: Dictionary
7.	IndFreq: Dictionary
8.	NofDocs: Integer
9.	Indf: Integer
10.	Individual: String
11.	Idf: Float
12.	If_Idf_Pairs: Dictionary
13.	**Output**:
	BEGIN
14.	NofDocs = COUNT(Documents)
15.	FOR a Document D IN Documents //compute DocFreq for each term/individual from documents
16.	UniqueIndividualsInD.ADD (FINDUNIQUEINDIVIDUALS (D,"\[[^<]+\]<.*?>")) //find all terms(i.e. individuals) that matches with the pattern '\[[^<]+\]<.*?>' from document D and add them to UniqueIndividualsInD list
17.	FOR Individual I IN UniqueIndividualsInD
18.	DocFreq[I] +=1
19.	Next
20.	Next
21.	FOR a Document D IN Documents //compute tf_idf
22.	AllIndividualsInD.ADD (FINDALLINDIVIDUALS (D,"\[[^<]+\]<.*?>")) //find all individuals that matches with the pattern '\[[^<]+\]<.*?>' from document D and add them to AllIndividualsInD
23.	FOR Individual I IN AllIndividualsInD
24.	IndFreq [i] +=1

25.	Next
26.	FOR I IN IndFreq
27.	Indf= I[0]
28.	Individual = I[1]
29.	Idf = ndocs/docFreq[Individual]
30.	If_idf_pairs +=[Indf *log10(idf), individual]
31.	Next
32.	Next
33.	RETURN If_Idf_Pairs
34.	**END**

Algorithm 4-7 is used to compute weight for concepts in a document.

Algorithm 4-7: Concept Weighting Algorithm

Line	**Concept Weighting Algorithm:**
1.	**Input**:
2.	Documents: List of **"Stemmed and Tagged"** documents
3.	O: Ontology
4.	**Variables**:
5.	AllConceptsInOntology: List
6.	AllIndConPairsInD: Dictionary
7.	UniqueConceptsInD: List
8.	AllConceptsInD: List
9.	DocFreq: Dictionary
10.	ConFreq: Dictionary
11.	NofDocs: Integer
12.	Conf: Integer
13.	Idf: Float
14.	Cf_idf: Float
15.	Cf_Idf_Pairs: Dictionary
16.	Concept: String
17.	**Output**:
	BEGIN
18.	READ Concepts FROM O //loading all the concepts from the ontology
19.	FOR each Concept C IN Concepts

```
    //load  all concepts from the ontology and convert them into geez
    form and apply stemming on them
20.     GeezifiedConcept = GEEZIFY (I)
21.     StemmedConcept = STEMM(GeezifiedConcept)
22.     AllConceptsInOntology.ADD(GeezifiedConcept)
23. FOR a Document D IN Documents
    //compute DocFreq for each concept from documents
24.     AllIndConPairsInD.ADD (FINDUNIQUEINDICONPAIRS (D))
        //find all individuals of each concept in the document D and add
        them to AllIndConPairsInD
25.     FOR a Concept C IN AllConceptsInOntology
26.        IF C is IN D
27.           DocFreq[C] +=1
28.        ELSE
29.           FOR each ind-con-pair IN AllIndConPairsInD
30.              IF ind-con-pair.READVALUE == C
31.                 DocFreq[C] +=1
32.           Next
33. Next
34. FOR a Document D IN Documents
    //compute tf_idf for each concept in each document
35.     AllIndConPairsInD.ADD (FINDUNIQUEINDICONPAIRS (D))
        //find all individuals of each concept in the document D and add
        them to AllIndConPairsInD
36.     FOR each ind-con-pair IN AllIndConPairsInD
37.         IF ind-con-pair.READVALUE == C
38.              ConFreq[C] +=1
39.     Next
40.     FOR a Concept C IN AllConceptsInOntology
41.        IF C is IN D
42.           AllConceptsInD.ADDALLOCUURENCES(C)
43.     Next
44.     FOR Concept C IN AllConceptsInD
45.        ConFreq[C] +=1
46.     Next
47.     FOR I IN ConFreq
48.        Conf= I[0]
```

```
49.             Concept = I[1]
50.             Idf = NofDocs/docFreq[Concept]
51.             Cf_idf = Conf*log10(idf)
52.             Cf_idf_pairs +=[Cf_idf, Concept]
53.         Next
54.     Next
55.     RETURN Cf_Idf_Pairs
56.     END
```

Ontology Population (OP) Module

In this module, the semantic information extracted by the IE module, the concept weights computed using the CW module, and the document URLs will be added in to the ontology. The main purpose of having this OP module is to associate documents with concepts in the ontology in order to increase the speed of the system and reduce storage space. By associating concepts with documents we enable the ontology to act as both a knowledge base and an index repository.

There are two main tasks in this module: -

1. **Adding the information extracted by the IE module into the ontology**

This task involves populating the ontology with information extracted from documents using the IE algorithm. The extracted information contains individuals and the relationships between them. The *Individual Adder algorithm*, Algorithm 4-1, developed in the ontology development component is used to add these individuals into the ontology. Moreover, the relationships between these individuals are added to the ontology using the *Relationship Creator Algorithm*, Algorithm 4-2, developed during ontology development.

In the example given in the IE module, Match ("ሐዋሳ ከነማ vs ሙገር ሲሚንቶ"), is a newly extracted instance/individual of type "ጨዋታ". This new concept has "1" value for its "Team1's Point" and "0" value for its "Team2's point" Datatype properties. Using *Individual Adder algorithm*, Algorithm 4-1***Error! Reference source not found.***, the "ሐዋሳ ከነማ vs ሙገር ሲሚንቶ" instance is created for the concept "ጨዋታ" and the values for the two Datatype properties are added for the newly created instance. Moreover, relationships between this instance and the other concept

instance (“16ኛው ሳምንት”, “ኢትዮጵያ”, “ፕሪሚየር ሊግ”, “ሐዋሳ”, “ሐዋሳ ከነማ”, “ሙገር ሲሚንቶ”) are created using the *Relationship Creator Algorithm*, Algorithm 4-2.

2. Adding those weights calculated by the CW module for concepts and individuals

This task is intended to populate the ontology with document links/URLs and weights (i.e. the concept and individual weights computed by the CW module) using Algorithm 4-1***Error! Reference source not found.*** (the *Individual Adder algorithm)* and Algorithm 4-2 *(the Relationship Creator algorithm)*. The *Individual Adder algorithm* creates an instance for the concepts “weight” and “Links” and the *Relationship Creator algorithm* creates relationship between instances of these concepts using the object properties “Has_Weight” and “Has_Link”.

4.4 Index structure

As it is discussed in Section 1.2, the main purpose of the study is to provide an index structure which has both the knowledge base and the index in one place so that the system will be able to respond to most of the queries with less time. This is accomplished by adding two classes to the ontology – Weight and Link. Every concept is connected with the “Weight” class with the object property “has_weight” and also with the class “Link” with the object property “has_link”. The classes Weight and Link are added at the top of the hierarchy as shown in Figure 4-5.

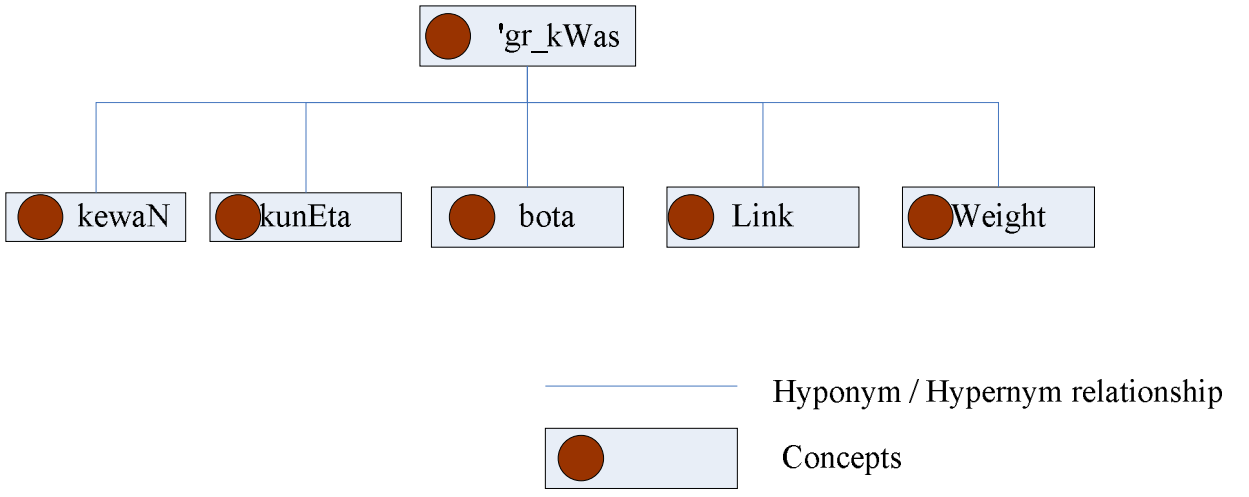


Figure 4-5: The high level ontology structure with the Link and Weight classes - an Index

Once the ontology is populated with concept weights and document URLs, it is taken as an index. Table 4-2 shows sample retrieved data from the populated ontology (the index).

Table 4-2: Sample Index Terms Retrieved From the Index

Index term[7]	Link(Document URL)	Weight
AseltaN	E:\thesis\OntobasedSemanticIndexer\news to be indexed\sportNews34.txt	0.2145213
Dedebit	E:\thesis\OntobasedSemanticIndexer\news to be indexed\sportNews123.txt	0.3456768

4.5 Query Processor

The design and implementation process of the semantic indexer has been done so far. Furthermore, this indexer has to be validated through a well defined experiment. Therefore, an information retrieval (IR) system, which utilizes the proposed semantic indexer, has been developed. The query processor component is responsible for designing and implementing this information retrieval system. Query processing involves the activities of retrieving and ranking relevant documents. This component uses the preprocessing and the concept tagging modules from the indexing component for query preparation and tagging. The main modules incorporated with this component are shown in Figure 4-6 and discussed in detail as follows.

[7] Note that Index terms are the whole concepts and individuals in the ontology

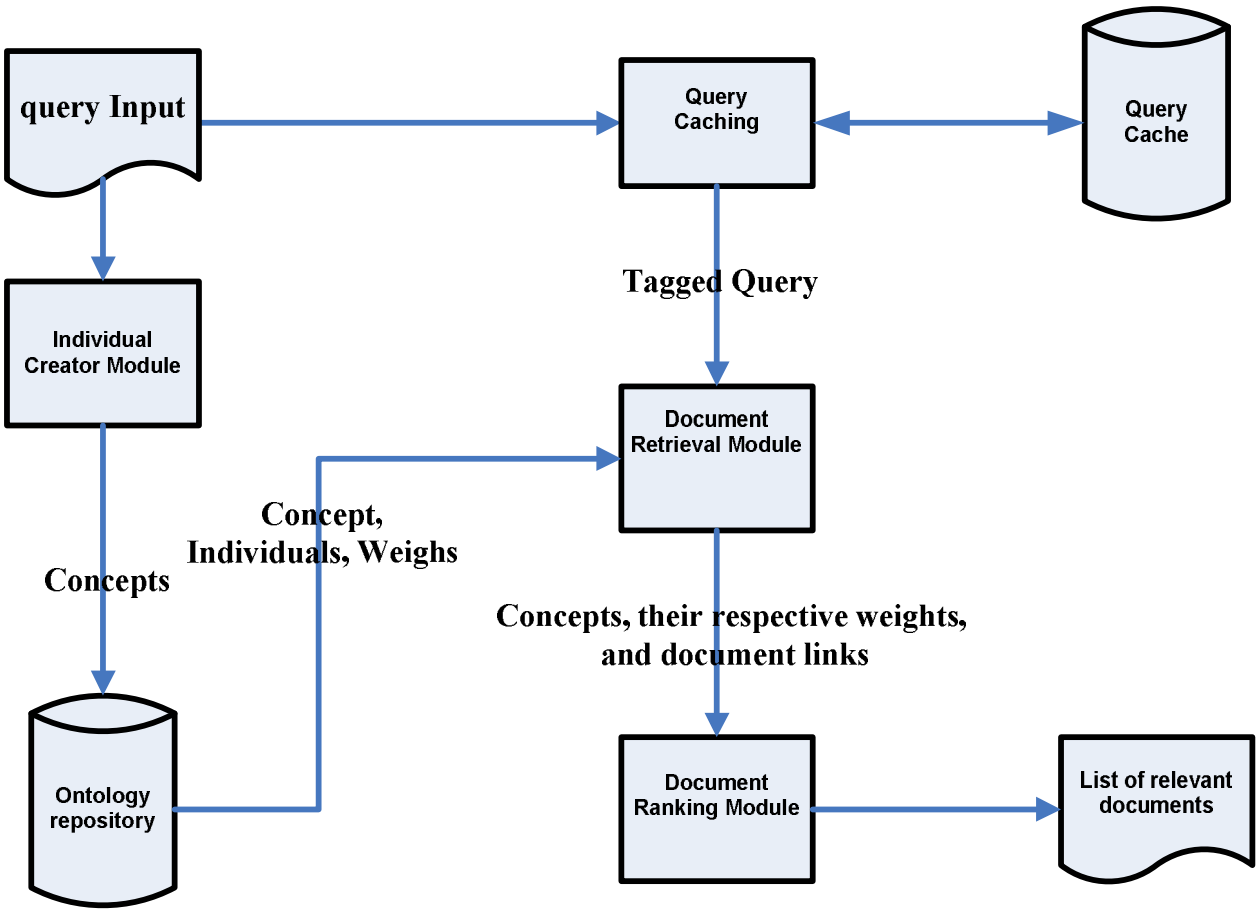


Figure 4-6: The interaction among the modules incorporated in query processing

Query Caching Module (QC)

Query caching is a process of registering queries in a query cache so as to make query processing faster. Query cache is a repository where set of query pairs are stored. A query pair is composed of an original query in a plain text and its semantically tagged equivalent. The original query is the initial query/text given by the user and the tagged query is the text that comes as a result of preprocessing and concept tagging processes. Whenever a query is posed, the system checks if the query exists in the query cache or not. When a query is found in the cache, the equivalent preprocessed and tagged query will be retrieved and passed to the next module, Document retrieval module (DR module). On the contrary, when a query is not found in a query cache (i.e. a new query) then preprocessing and concept tagging tasks will be applied on it and a new query

pair will be formed and inserted into the query cache. The preprocessing and the concept tagging activities are done by the preprocessing and concept tagging modules built in the preprocessor component. A query cache is designed by considering the probability that users may write similar queries to get information out of documents. Hence, registering queries and their tagged version helps to minimize the time needed to process a query. Algorithm 4-8 is used to register queries in the query cache.

Algorithm 4-8: Query Caching Algorithm

Line	**Query Caching Algorithm:**
1.	**Input**:
2.	Q: Query
3.	QueryCache: Query repository
4.	QueryPairs: List of queries from the Query Cache - (Q,TQ) Q –Original Query, TQ is Tagged Query
5.	**Variables**:
6.	TaggedQuery: String
7.	NormalizedQuery: String
8.	StemmedQuery: String
9.	**Output**:
	BEGIN
10.	READ Individuals FROM O
11.	FOR each QueryPair QP IN QueryPairs
12.	IF Q is IN QP
13.	READ TQ ///read the tagged version of Q and pass it to the Document Retrieval Module
14.	TaggedQuery = QP
15.	Next
16.	IF TaggedQuery == *“”* // if the original query is not found in the query chache
17.	NormalizedQuery = NORMALIZE(Q) // using the preprocessing module from the indexer component
18.	StetmmedQuery = STEMM(NormalizedQuery) // using the preprocessing module from the indexer component
19.	TaggedQuery = TAGG(StetmmedQuery) //tagging individuals using the *“Tagging Algorithm2”* from the

	indexer component
20.	QueryCache.ADD(Q, TaggedQuery) //adding a new query pair into the query cache
21.	RETURN TaggedQuery, QueryCache
22.	**END**

The following scenario shows how the Query Caching Algorithm, Algorithm 4-8, works:

Original Query – Query1: "የ2004 የኢትዮጵያ ፕሪምየር ሊግ ዕግር ኳስ ውድድር ወሸናፊ ቡድን" is given as an input to the Query Caching algorithm

It checks if the Query exists in the Query Cache and returns the tagged version of the query

Otherwise: - it normalizes, stems, and tags the Original Query as "[2004] <ዘመን> [ኢትዮጵይ]<ሀገር> [ፕሪምየር ሊግ]<ሊግ> እግር ኳስ ውድድር አሸናፍ ቡድን" and adds the original query and the tagged version to the query cache.

Individual Creator Module (IC module)

The ontology is constructed using almost all available concepts in the football domain and the probability that new concepts can be created in this particular domain is believed to be less. On the other hand, new concept instances/individuals can be produced often. For example, new football clubs can be established at any time and these newly formed clubs will be taken as instances of the concept "ቡድን". Therefore, there is a probability that a certain query may not have any recognized concept or individual by the ontology and due to this reason the system will be incapable of providing documents with possible answers for such kind of queries. In such cases, relevant terms should be extracted from the query and added into the ontology as individuals so that when next time the same query is posed the system will be able to use these individuals and respond to the query. This module, IC, is dedicated to accomplish this task.

IC takes a query as an input, produces individuals from the query, and adds them into the ontology. In order to make use of the newly added individuals, re-indexing of documents needs to be done within a certain period of time, i.e., indexing schedule should be set. According to the

authors in [49], significant multi-word terms, in a corpus, are most probably to be sequences of nouns. The authors proved that using only noun sequences to produce multi-word terms provides high precision. Similarly, in our study, we considered the consecutive nouns that are in the query as relevant multi-word terms. Besides, the nouns that appear alone are taken as relevant single terms of the query. As it is difficult to identify the exact class of all the individuals extracted from queries, all of the extracted terms are added into the ontology as instances of the concept "እግር ኳስ". The concept "እግር ኳስ" is chosen due to the fact that every individual of the concepts in the ontology are also individuals of the concept "እግር ኳስ" because "እግር ኳስ" is found at the top of the hierarchy.

In order to identify nouns in the query, the Part Of Speech (POS) tagger built along with the hornMorpho software developed by Gasser [47] is used. For example, if the query "የ ገመና ድራማ አርቲስቶች ከ ባለስልጣናት" has no recognized concept or individual by the concept tagger module, then the POS tagger annotates the query as "የገመና[N] ድራማ[N] አርቲስቶች[N] እና ባለስልጣናት[N]". Tow concept instances/individuals–"የገመና[N] ድራማ[N] አርቲስቶች[N] "and "ባለስልጣናት[N]' are extracted and added into the ontology. This task is performed by applying Algorithm 4-9.

Algorithm 4-9: Individual Creator Algorithm

Line	**Individual Creator Algorithm:**
1.	**Input**:
2.	Query: Normalized query
3.	O: Ontology
4.	**Variables**:
5.	ListofNouns: List
6.	RomanizedNoun: String
7.	**Output**:
	BEGIN
8.	taggedQuery = ANNOTATE(Query)) //annotating the query with the POS tagger – identifying nouns from the query
9.	ListofNouns.ADD(EXTRACTCONSNOUNS(taggedQuery, Nouns)) // extract nouns from the query and add them into the list ListofNouns
10.	FOR each noun N IN ListofNouns

11.	RomanizedNoun = ROMANIZE(N) //changing the nouns from geez to roman e.g. “ባለስልጣናት” to “balslTanat”
12.	O.CREATEINDIVIDUAL(“'gr_kWas”, RomanizedNoun) //create an individual for the concept “'gr_kWas” with the name N
13.	Next
14.	RETURN O
15.	**END**

Document Retrieval module (DR module)

The role of the DR module is to find the set of relevant documents for a particular query using the tagged version of the original query. During the concept tagging process, only individuals are identified, i.e, only individuals are annotated in the tagged query. For example, referring to the tagged query, “[2004] <ዘመን> [ኢትዮጵይ]<ሀገር> [ፕሪምየር ሊግ]<ሊግ> እግር ኳስ ውድድር አሸናፍ ቡድን”, it contains only individuals “2004”, “ኢትዮጵይ”, and”ፕሪምየር ሊግ”. In fact, besides these individuals, the concepts (“እግር ኳስ”, “ውድድር”, and “ቡድን”) should also be extracted from the query. Therefore, concept extraction is done using the concept identification part of the ‘Concept Weighting’ algorithm – developed in the indexer component. The concepts and individuals in a query are represented as follows.

Q_n (C_1, C_2…., I_1, I_2…), where {n is the query number like - $Query_1$,

C is a concept in the query Qn, and

I is an individual in the query Qn}

For example, the above query is represented as: Q_1(“2004”, “ኢትዮጵይ”,”ፕሪምየር ሊግ”, “እግር ኳስ”, “ውድድር”, “ቡድን”).

Document retrieval process involves two main activities:-

- Inferencing - Concept reasoning
- Retrieving weights and document links

Inferencing - Concept reasoning

Inferencing is the process of identifying concepts that have direct or indirect relationship with the concepts in the query and computing concept similarity. Inferencing, using related concepts, helps to find relevant documents for indirect queries, i.e., to respond to queries even though they do not share a single concept with any document. The relevance that an inferred concept has to a query is important to determine the order of relevance of retrieved documents. Concept relevance is calculated using the distance between inferred and original concepts – concepts in the query. There are different built in reasoners in Jena but they lack the capability of providing the distance between concepts – no method is available to measure concept similarity. Therefore, it is found to be vital to design and develop a separate concept reasoner so that the distance between inferred concepts and original concepts can be computed. The concept distance will be used later in document ranking.

Inferencing is done for each of the individuals and the concepts in the Query. The inferred items for a concept C in a particular query Q using the resoner built in this study are: -

- Individuals of C,
- All direct and indirect Super Concepts,
- All direct and indirect sub Concepts
- All siblings of C – concepts which share same direct super concept with C.

Whereas the inferred items for an individual I in a particular query Q are: -

- The type/class of I – the concept C in which this individual I belongs,
- All direct and indirect Super Concepts of C,
- All direct and indirect sub Concepts of C
- All siblings of I - those individuals that share same class with I – all individuals of C

For example, for an original query – Query2 (“በ16ኛው ሳምንት የኢትዮጵያ ፕሪሚየር ሊግ የሐዋሳ ከነማ እና የሙገር ሲሚንቶ ጨዋታ ማን አሸነፉ”), which is represented as Q_2(“ሊግ”, “ጨዋት”,“ኢትዮጵይ”, “ፕሪሚያ ር ሊግ”, “16ኝ ሳ ምን ት”). The inferred items for both concepts and individuals in Query2 are: -

- For each concept (“ሊግ” and “ጨዋታ”) extract the following information:
  - Individuals

    “ሊግ”: - “ፕሪሚየር ሊግ”, “ብሄራዊ ሊግ “ …

    “ጨዋታ”: - “ሐዋሳ ከነማ vs ሙገር ሲሚንቶ”, “ደደቢት vs መከላከያ”, “ቅዱስ ጊዮርጊስ vs ደደቢት” …
  - Their entire super concepts

    “ሊግ ”: -“ውድድር ”, “ሁኔ ታ”, and “እ ግ ር ኳስ ”

    “ጨዋ ታ”: -“ሁኔ ታ” and “እ ግ ር ኳስ ”
  - Their entire sub concepts.

    “ሊግ ”: it has no sub concepts

    “ጨዋ ታ”: it has no sub concepts
  - Siblings

    “ሊግ”: -“ዋንጫ”

    “ጨዋታ”: -“ውድድር”, “ሽልማት”, “ቆይታ”, “ዘመን”. ”የተጫዋች ሁነታ”, “የ ዳኛ ሁነታ”, and “ዙር ”
- For each individual (“ኢትዮጵያ”, “ፕሪሚየር ሊግ”, and “16ኛው ሳምንት”)extract the following information:
  - Individual Type

    “ኢትዮጵያ”: - “ሀገር”

    “ፕሪሚየር ሊግ”: - “ሊግ”

    “16ኛው ሳምንት”: -“የ ፕሪሚየር ሊግ ዙር”
  - Super concepts of the Individual type “ሀገር”,“ሊግ” and “የ ፕሪሚየር ሊግ ዙር”

    “ሀገር”:-“አህጉር”,”ከዋኝ”, and “እግር ኳስ”

    “ሊግ”:-“ውድድር”, “ሁኔታ”, and “እግር ኳስ”

    “የ ፕሪሚየር ሊግ ዙር”:-“ዙር”, “ሁኔታ”, and “እግር ኳስ”
  - Sub concepts of the Individual type “ሀገር”,“ሊግ” and “የ ፕሪሚየር ሊግ ዙር”

    “ሀገር”:-it has no sub concepts

    “ሊግ”:-it has no sub concepts

    “የ ፕሪሚየር ሊግ ዙር”:- it has no sub concepts
  - Siblings

    “ኢትዮጵያ”: - “ኬንያ”, “ግብፅ”, “ዛምቢያ”....

    “ፕሪሚየር ሊግ”: - “ብሄራዊ ሊግ” …

    “16ኛው ሳምንት”: - “1ኛው ሳምንት”, “2ኛው ሳምንት”...

The relevance of each of the inferred items (concepts and individuals) to a query is determined by the distance the items have to the original concept or individual. The similarity/distance between *inferred* and *original concepts* of a query Q is shown in Table 4-3. Similarly, the similarity between inferred items and original individuals of a query Q is shown in Table 4-4.

Table 4-3: The similarity between inferred items and original concept C in a query Q

	Inferred items using C	**The similarity of the inferred item to the original concept C**	**Remark**
Original Concept C (a concept C in Q)	Individuals	0.75	The relevance of all instances to the query Q is 0.75
	Super concepts	0.5/2	The relevance of the direct parent of the concept is 0.5 and as we go from this parent concept to the top of the hierarchy the relevance of the concepts decreases by half
	Sub concepts	0.5/2	The relevance of the direct sub concept of the concept C is 0.5 and as we go from this parent concept to the bottom of the hierarchy the relevance of the concepts decreases by half
	Siblings	0.5	The relevance of all siblings of the Concept C to the query Q is 0.5

Table 4-4: The similarity between inferred items and original Individual C in a query Q

	Inferred items from I	**The similarity of the inferred item to the original individual I**	**Remark**
the original Individual I(an Individual I in Q)	Individual type (the class that I belongs to)	0.75	The relevance of the Individual type (IT) to the query Q is 0.75
	Siblings	0.5	The relevance of all siblings of the Individual I to the query Q is 0.5
	Super concepts of the individual type (IT)	0.5/2	The relevance of the direct parent of IT is 0.5 and as we go from this parent concept to the top of the hierarchy the relevance of the concepts decreases by half
	Sub concepts of the individual type (IT)	0.5/2	The relevance of the direct sub concept of IT is 0.5 and as we go from this parent concept to the bottom of the hierarchy the relevance of the concepts decreases by half

The following algorithm, Algorithm 4-10 , is used to build the reasoner which computes inferencing with similarity values.

Algorithm 4-10: Inferencing/Concept Reasoning Algorithm

Line	**Inferencing/Concept Reasoning Algorithm:**
1.	**Input**:
2.	OriginalConcepts: List of original concept from Q
3.	OriginalIndividuals : List of original individuals from Q

4.	O : Ontology
5.	**Variables**:
6.	RomanizeConcept: String
7.	RomanizedIndividual : String
8.	AllSuperConceptsofC : List
9.	AllSubConceptsofC: List
10.	ALLIndividualsofC = List
11.	AllSibilingsofC = List
12.	Relevance: Float
13.	TermRelevancePairs: Dictionary //term denotes both concepts and individuals
14.	AllSuperConceptsofI: List
15.	AllSubConceptsofI: List
16.	AllSibilingsofI = List
17.	IndividualTypeofI = List //the class or concept I belongs to
18.	**Output**:
	BEGIN
19.	FOR each Concept C IN OriginalConcepts // infer items based on concepts in the query and compute similarity/relevance
20.	RomanizeConcept = ROMANIZE (C)
21.	TermRelevancePairs.ADD(RomanizeConcept, 1)
22.	AllSuperConceptsofC.ADD(O.GETSUPERCONS(RomanizeConcept))
23.	AllSubConceptsofC.ADD(O.GETSUBCONS(RomanizeConcept))
24.	ALLIndividualsofC.ADD(O.GETINDIVIDUALS(RomanizeConcept))
25.	AllSibilingsofC.ADD(O.GETSIBILINGD(RomanizeConcept))
26.	Next
27.	Relevance = 1
28.	FOR a concept C AllSuperConceptsofC
29.	Relevance = Relevance/2
30.	TermRelevancePairs.ADD(C,Relevance)
31.	Next
32.	Relevance =1
33.	FOR a concept C AllSubConceptsofC
34.	Relevance = Relevance/2
35.	TermRelevancePairs.ADD(C,Relevance)

```
36.     Next
37.     FOR a concept C ALLIndividualsofC
38.         Relevance = 0.75
39.         TermRelevancePairs.ADD(C,Relevance)
40.     Next
41.     FOR a concept C AllSibilingsofC
42.         Relevance = 0.5
43.         TermRelevancePairs.ADD(C,Relevance)
44.     Next
45.     FOR each Individual I IN OriginalIndividuals
        // infer items based on individuals in the query and compute
        similarity/relevance
46.         RomanizeIndividual = ROMANIZE (I)
47.         IndividualTypeofI = O.READINDIVIDUALTYPE(RomanizeIndividual)
48.         TermRelevancePairs.ADD(RomanizeConcept, 1)
49.         TermRelevancePairs.ADD(IndividualTypeofI, 0.75)
50.         AllSuperConceptsofI.ADD(O.GETSUPERCONS(IndividualTypeofI))
51.         AllSubConceptsofI.ADD(O.GETSUBCONS(IndividualTypeofI))
52.         AllSibilingsofI.ADD(O.GETSIBILINGD(RomanizeIndividual))
53.     Next
54.     Relevance = 1
55.     FOR each Individual I IN AllSuperConceptsofI
56.         Relevance = Relevance/2
57.         TermRelevancePairs.ADD(I,Relevance)
58.     Next
59.     Relevance = 1
60.     FOR each Individual I IN AllSubConceptsofI
61.         Relevance = Relevance/2
62.         TermRelevancePairs.ADD(I,Relevance)
63.     Next
64.     FOR each Individual I IN AllSibilingsofI
65.         Relevance = 0.5
66.         TermRelevancePairs.ADD(I,Relevance)
67.     Next
68.     RETURN TermRelevancePairs
69.     END
```

Retrieving weights and document links

In the second activity of this module, weights and document links for all the original and inferred items (concepts and individuals) from the ontology (the index) are retrieved. Document link/URL is an absolute path to a document. The results of this activity and the previous one are passed to the Ranking module for the purpose of document ranking i.e. the items (original and inferred concepts and individuals), their corresponding relevance value (similarity value), and weight-document pairs are inputs to the ranking module. Weight-document pair combines the weight an item (concept or individual) has in a document and the document link/path/URL.

For example, for a query Q_2("ሊግ", "ጨዋት", "ኢትዮጵይ", "ፕሪሚየር ሊግ", "16ኛ ሳምንት", "እግር ኳስ"), the data shown in Table 4-5 is given to the ranking module so as to sort the relevant documents to the query Q_2

Table 4-5: Sample retrieved documents using the DR module

Query Item(Concept or Individual)	Relevance	Item Weight	Document Link (URL)
"ሊግ"	1	0.2344445	E:\thesis\OntobasedSemanticIndexer\news to be indexed\sportNews0.txt
		0.1346445	E:\thesis\OntobasedSemanticIndexer\news to be indexed\sportNews2.txt
…..	…..	……	…..
"እግር ኳስ"	0.125	0.019344	E:\thesis\OntobasedSemanticIndexer\news to be indexed\sportNews0.txt
		0.12345	E:\thesis\OntobasedSemanticIndexer\news to be indexed\sportNews12.txt

If documents that are retrieved using the inferred items (concepts and individuals) also occur in those documents retrieved using an original item, then the one retrieved using the inferred item is removed from the relevant document collection for the sake of eliminating redundancy. For

instance, in the example above, the item "እግር ኳስ" is an inferred item from the item "ሊግ" and they both occur in the document "E:\thesis\OntobasedSemanticIndexer\news to be indexed\sportNews0.txt". In this case, the document which is retrieved using the inferred item "እግር ኳስ" should be removed along with its weight value("0.019344") because "እግር ኳስ" is less relevant to the query than "ሊግ" is since it does not exist explicitly in the query (i.e., the 3rd entry should be removed from the list).

Document Ranking Module

The document ranking module is dedicated to sort the retrieved documents according to their significance to a query. The total weight that an item (original/inferred concept/individual) has in a specific document is computed using the formula in Equation 4-4.

$$\textbf{Total Weight}_i = \textbf{Relevance}_i \times \textbf{Weight}_i \quad [\textbf{4-4}]$$

Where

{$Relevance_i$ is the similarity value that item i has with the original item, and

$Weight_i$ is the item i's (concept or individual) weight stored in the index -ontology}

For instance, using the example given in the Document retrieval module, the Total weight is computed as shown in Table 4-6.

Table 4-6: Sample ranked documents using the Docuemtn Ranking module

Concept/Individual	Total Weight	Document link/ URL
“ሊግ”	1*0.2344445 = 0.2344445	E:\thesis\OntobasedSemanticIndexer\news to be indexed\sportNews0.txt
	1* 0.1346445 = 0.1346445	E:\thesis\OntobasedSemanticIndexer\news to be indexed\sportNews2.txt
…	…	…
“እግር ኳስ”	0.125*0.019344 = 0.0024175	E:\thesis\OntobasedSemanticIndexer\news to be indexed\sportNews0.txt
	0.125*0.12345 = 0.01543125	E:\thesis\OntobasedSemanticIndexer\news to be indexed\sportNews12.txt

If there are more than one number of items in a query – Q(i1, i2, i3, …), then those documents which have most of the items are given higher rank, i.e, a document which has most of the concepts/individuals from the query is most relevant to the query than other documents. For example, if a query Q has 3 items (either concepts or individuals or both) and if there is a document D1 containing all of the 3 items from Q and a document D2 has only two of the items, then D1 will be given higher rank. In addition, if more than one document have equal number of items from the query, then they will be ranked according to their total weight value – the one with highest total weight will come first and it will in descending order up to the last document. The following algorithm,

Algorithm 4-11 is used by the ranking module to get the final set of ranked documents.

Algorithm 4-11: Ranking Algorithm

Line	**Ranking Algorithm:**
1.	**Input**:
2.	ItemList: List of items with relevance value, weight, and document link as {item, relevance, weight, docLink}

```
3.    Variables:
4.    TotalWieghtDocListPairs: List
5.    TotalWeight: Float
6.    MaxFrq: Integer
7.    DocsWithEqualFreq: List
8.    DocsWithEqualFreq_TotWePairs: List
9.    SortedDocs: List
10.   AllDocs:List
11.   Output:
      BEGIN
12.   FOR an entry E IN ItemList
13.     READ Item, Relevance, Weight, DocLink FROM E
14.     TotalWeight = Relevance*Weight
15.     TotalWieghtDocListPairs.ADD(TotalWeight,DocLink)
16.     AllDocs.ADD(DocLink)
17.   Next
18.   MaxFrq = COUNTDOC(AllDocs)
      // get the frequency of the document which occurs most
      repetitively than others – the document which has most of the
      items in the query
19.  FOR Integer I= MaxFreq, I>0; I--
20.       FOR doc IN  AllDocs
21.            IF COUNTDOC(Doc) is = I THEN
22.                DocsWithEqualFreq.ADD(Doc)
23.       Next
24.       FOR Doc IN DocsWithEqualFreq
25.            READ TotalWeight FROM TotalWieghtDocListPairs WHERE
               DocLink = Doc
26.            DocsWithEqualFreq_TotWePairs.ADD(TotalWeight, Doc)
27.       Next
28.       SortedDocs.ADD(SORT(DocsWithEqualFreqTotWePairs,
          "TotalWeight", "Descending"))
          //sort the documents with equal frequency using their
          TotalWeight value in descending order
29.   Next
30.   RETURN SortedDocs
31.   END
```

Chapter Five - Experiment and Evaluation

5.1 Overview

As mentioned in Chapter 3, an information retrieval system has been developed with the intention of using the proposed indexer in a specific application domain so that its performance can be tested. Thus, in this chapter, the set of experiments conducted to validate the proposed indexing approach are presented.

We have followed some set of procedures to conduct the experiment. The test environment, the set of activities defined under the procedures, and the findings from the experiment are described in detail in the following sub sections.

5.2 Experimental Procedure

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

5.2.1 Data/Test set Collection

Three sets of data were required in order to undertake the experiment. The first set of data is a collection of real football concept instances referred to as *"concept instance data set"*. This data is used to populate the ontology so that it can be used as a knowledge base. The source for the data was Ethiopian Football Federation (EFF). EFF has all the information necessary to populate the ontology such as names of all the players, the clubs, the referees, and so on. A questionnaire was prepared and given to the concerned person in EFF to provide us the required information in either hard copy or soft copy. The questionere is attached as Annex F.

The second set of data is a collection of Amharic football news documents from WIC information Center and this data set is referred to as "*news document data set*". 138 news documents were selected and prepared as test data for the query processing part (information retrieval system).

The third Set of data is a list of *Amharic football queries* gathered from people who frequently read football news from different newspapers and watch football game. 25 sample queries out of all the collected queries were selected and used for testing. 8 of the 25 queries (Query 8, Query 19 and Query 20 up to 25) are indirect and the rest are direct queries. These queries are attached as Annex G.

5.2.2 Manual Query Processing

Manual query processing is a way of providing relevant document list to each query based on the document collection. This process is called as manual because it is done by subject experts instead of the system and the output of this process is relevance information containing the lists of relevant documents returned by these experts to each query. The relevance information made for all the 25 queries using the 138 news articles are attached as Annex G. As an example, relevance documents for Query 18 are presented below in Table 5-1.

Table 5-1: Sample relevance information

Query Number	Query	Relevance Documents
Query 18	በ አፍሪካ ዋንጫ ኢትዮጵያ እና የቺፖሎፖሎዎች የመጀመርያ ግጥሚያ	sportNews134, sportNews135, sportNews133, sportNews128

5.3 Evaluation

The In order to validate the relevance of our concept based indexer, we have implemented the classical Amharic information retrieval system [2] which uses the lucene indexer. Recall, precision, and F-measure are used to evaluate the performance of these two systems. Recall is the ratio of the number of documents retrieved correctly to the total number of relevant documents in the document collection whereas precision is the ratio of the number of documents retrieved correctly to the total number of documents retrieved. F-measure is the standard measure for evaluating IR by combining recall and precision techniques.

In this study, all these three evaluation techniques are applied because of their considerable advantages. Precision and recall are used to show how many of the relevant documents are captured and missed by the proposed and the classical IR system for each query whereas the F-

measure shows the overall performance of the system for each query by combining the recall and precision values. The harmonic F-measure, which gives equal weight for recall and precision, is used in this study even though it is possible to give different weights. The harmonic F-measure is calculated using the formula in Equation [5-1.

$$F - measure = 2 * \frac{(Recall * Percision)}{(Recall + Percision)} \quad \textbf{[5-1]}$$

In most researches, expert judgment is considered to be correct and absolute. However, in this research we argue that the initial expert judgments are very limited and we advised experts to refine their query result judgment seeing the result of the two query system (following the relevance feedback approach in IR). The refinement process involves making relevance judgment once again by users/experts on the results returned by both the proposed and the classical IR systems. In this task, experts go through the documents retrieved by both the traditional and the proposed system and determine whether these documents can be relevant to the respective queries or not (i.e., rechecking if documents returned by the systems and yet missed by the experts are relevant).

The results returned by both systems for all the queries are attached as Annex H. The initial relevance information is refined based on these results and those relevant documents missed by experts but returned by the proposed and the classical IR systems are shown in Table 5-2 and Table 5-3 respectively.

Table 5-2: List of documents identified by the proposed system and missed by experts

Query Number	Relevant documents based on the proposed system
Query 3	sportNews0.txt
Query 7	sportNews125.txt, sportNews126.txt, sportNews127.txt
Query 8	sportNews134.txt
Query 9	sportNews127.txt
Query 21	sportNews134.txt
Query 25	sportNews118.txt

Table 5-3: List of documents identified by classical IR system and missed by experts

Query Number	Relevant documents based on the classical IR system
Query 3	sportNews0.txt
Query 7	sportNews125.txt, sportNews126.txt, sportNews127.txt
Query 9	sportNews127.txt

As we can see from Table 5-2 and Table 5-3, the proposed system has captured 8 relevant documents for 6 queries and the classical IR has returned 5 documents for 3 queries. This indicates that the relevance judgment made by experts is limited and also the proposed system has returned more relevant documents than the classical IR. Hence, the entries from Table 5-2 and Table 5-3 are added to the initial relevance information to form refined relevance information.

Precision and recall values are computed for all the queries using both the initial and refined relevance information for both systems. The results computed using the initial relevance information is attached as Annex I and those calculated using the refined one is attached as Annex J.

In addition, the recall, precision and F-values are computed for each of the 25 queries and presented in Figure 5-1 and Figure 5-2. The computation is done before and after the refinement has been done by experts.

Figure 5-1 particularly shows precision value and OSIATED provide accurate value compared to the classical system.

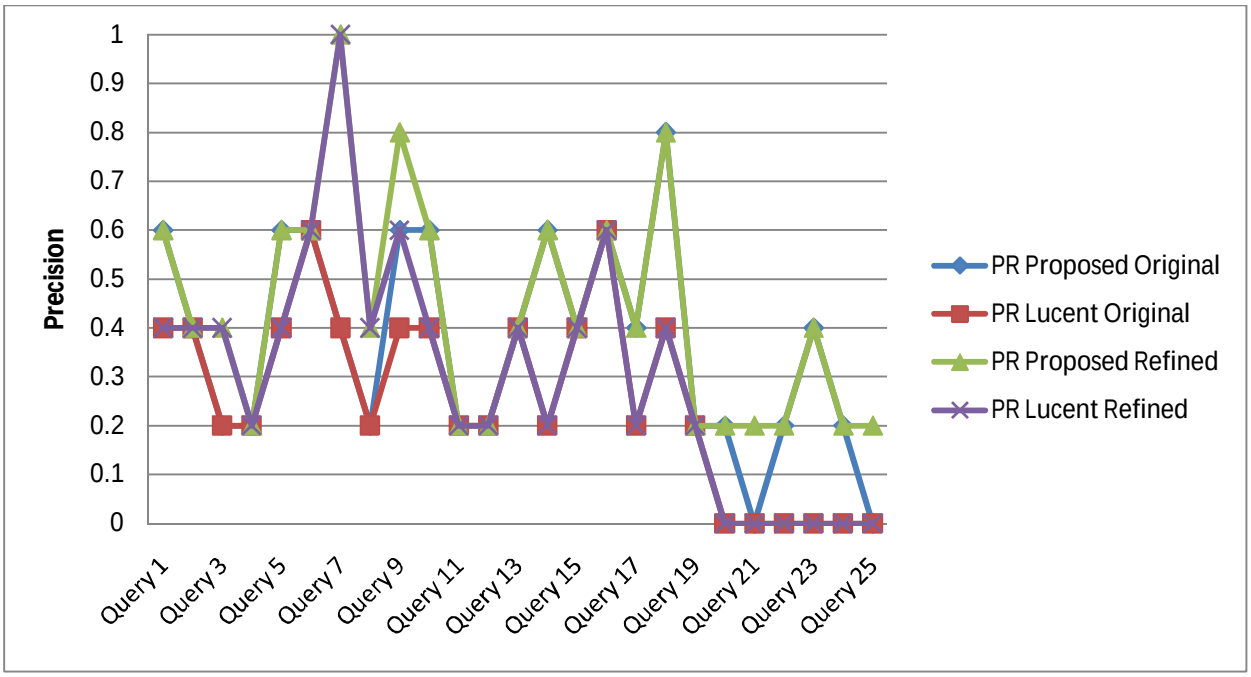


Figure 5-1: Precision Values Based On the Original and Refined Expert Judgment for All the Queries for Both the Proposed and the Classical IR (Lucent index)

When we compare the precision and recall values computed using the initial and refined relevance information for the proposed system by referring to Figure 5-1 and Figure 5-2, the precision has increased for "Query 3", "Query 7", "Query 8", "Query 9", "Query 21", and "Query 25" and the recall has increased for "Query 21" and "Query 25" when using the refined expert judgment. For example, when we look at "Query 25", it doesn't have any relevant document in the initial relevance information but when checking those documents returned by the proposed system, one relevant document, "sportNews118.txt", has been discovered and the precision for this particular query has increased from 0 to 0.2 and recall from 0 to 1. Like the proposed system, the classical IR has also captured those missed documents for the direct queries ("Query 3", "Query 7", and "Query 9") as shown in Figure 5-1. Thus, the precision for these 3 queries has increased for the classical IR as well. However, no change has been observed on recall for any of the queries for this system even though expert judgement refinement has been done. Besides, the classical IR has not captured any of the missed documents for the indirect queries. This shows that, unlike the classical IR, the proposed indexing method is based on semantics rather than simple key words.

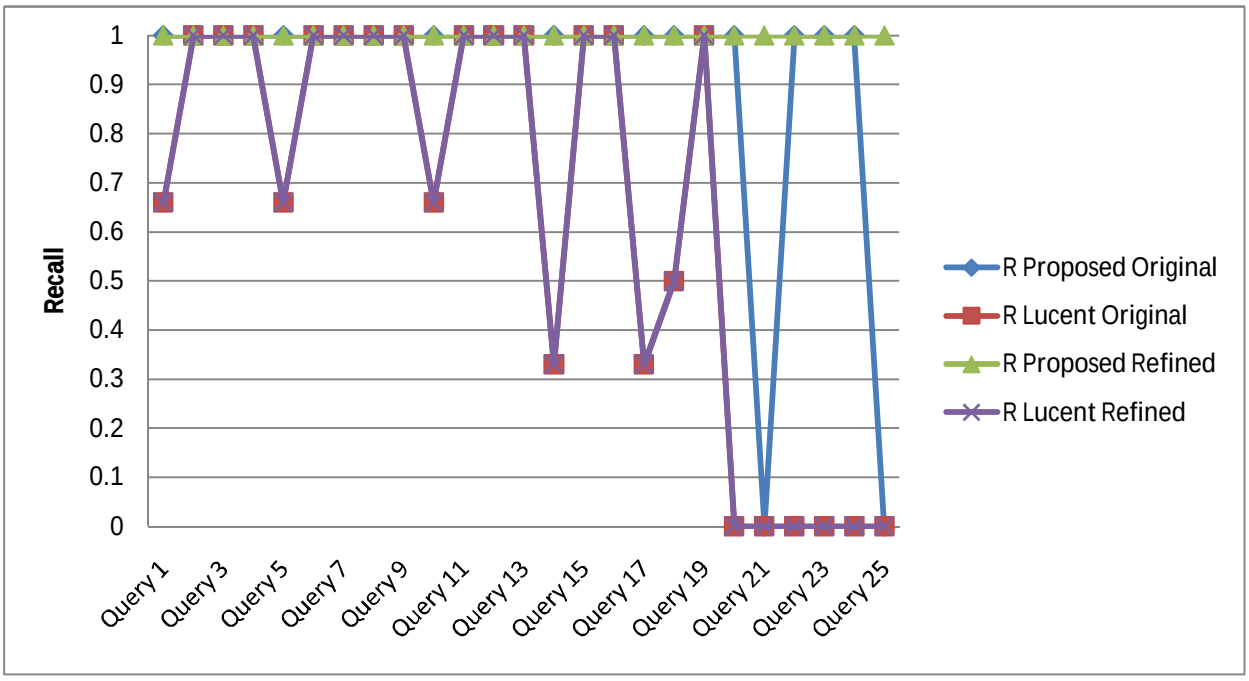


Figure 5-2: Recall Values Based On the Original and Refined Expert Judgment for All the Queries for Both the Proposed and the Classical IR (Lucent index)

Figure 5-2 shows that, the proposed system has a maximum recall value, 1, for all the queries even though it has returned some irrelevant documents. On the contrary, the classical IR system has missed some of the relevant documents for some queries and missed all for "Query 20" up to "Query 25". For example, "Query 23" – "የባፋና ባፋና አሰልጣኝ" has 2 relevant documents according to the refined relevance information but the classical IR has not returned any one of them because the term "ባፋና ባፋና" does not exist in these documents. In contrast, the proposed system captured the relevant document by referring to the fact that "ባፋና ባፋና" has same meaning with the term "የደቡብ አፍሪካ ብሄራዊ ቡድን".

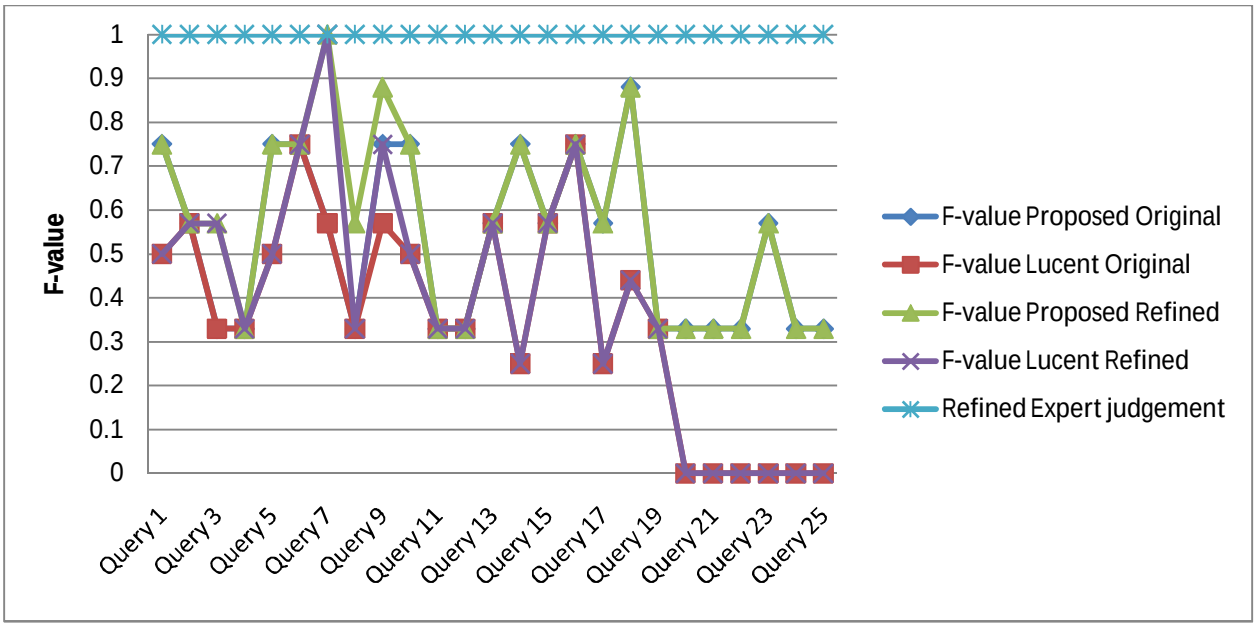


Figure 5-3: F-measure Values Computed Based On the Original and Refined Expert Judgment for All the Queries for Both the Proposed and the Classical IR

In order to show a better view of the capability of both systems, F-value has been computed for all queries of each of the systems (classical IR and the proposed systems). These F-measure values are plotted on a graph in Figure 5-3. Besides this, the F-measure values are also computed for the refined expert judgment and plotted on this graph. The data used to plot this graph is attached as Annex J.

Since the refined relevance judgment is believed to provide the exact answer for all the queries, the results of the proposed system and classical IR (lucent) are compared against the expert judgment. Figure 5-3 shows that the F-measure values for the proposed system for 14 queries are greater than that of the classical IR and equal for the rest of the queries. This indicates that for more than half of the queries, the result of the proposed system is much closer to the expert judgment as compared to the classic IR. Besides, the f measure values computed for the proposed system using the refined relevance information are greater than the values computed using the original/initial relevance information for some of the queries i.e. the “F-value Proposed Refined” line is much closer to the “Refined expert judgment” when compared to the “F-value Proposed Original”.

For the 25 user queries and set of documents retrieved using each system average recall, precision, and F-measure values have been identified and presented graphically in Figure 5-4. The average values were calculated over the number of queries. The F-measure values are computed with the intention of evaluating the result of the systems independent of the recall and precision values. For example, if two different queries have different recall and precision values but the summation of these values are similar then the F-measure values are similar as well.

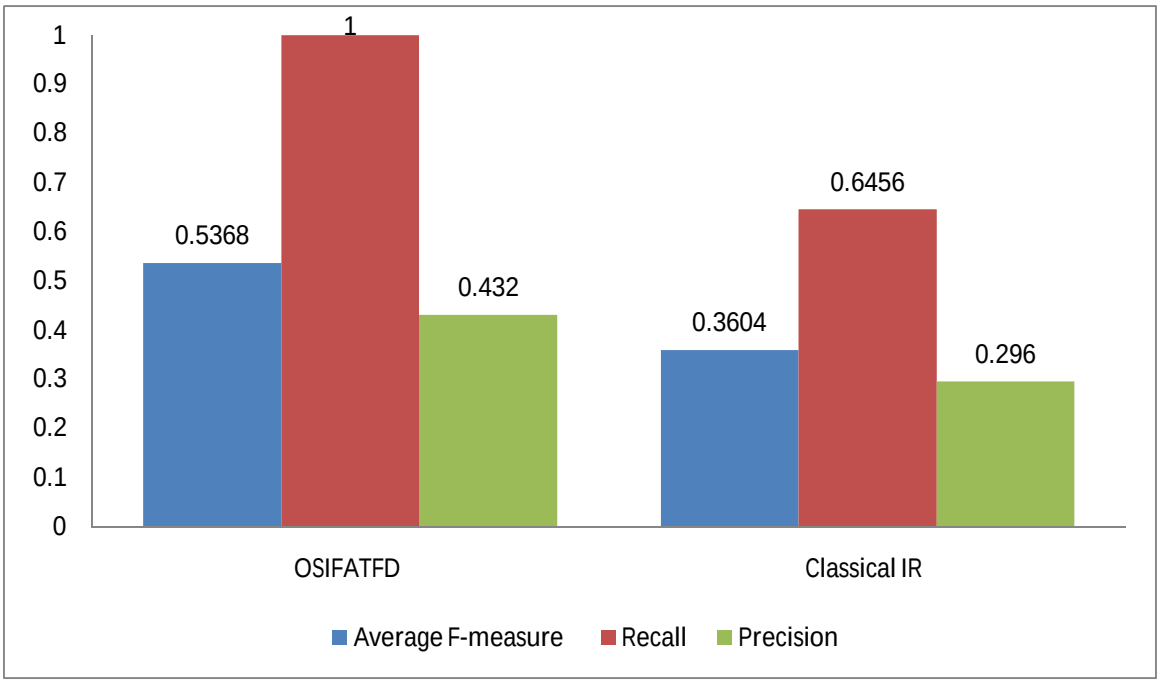


Figure 5-4: A comparison of the two systems used in the experiment

The data in Table 5-4 was used to generate the bar graph in Figure 5-4.

Table 5-4: Values used to plot Precision-recall bar graph for the two Systems

System	Recall (Average)	Precision (Average)	F-measure (Average)
Classical IR	0.6456	0.296	0.3604
Proposed System	1	0.432	0.5368

As it is illustrated in Figure 5-4, the average values for all the three evaluation techniques for the proposed system are greater than the respective values for the classical IR system. When we put the average values in a percentage, the proposed system has 100% recall, 43.2% precision, and

53.68% F-measure whereas classical IR has 64.56% recall, 29.6% precision, and 36.04% F-measure.

5.4 Discussion

Even though the recall of the proposed system is 100%, the precision is 43.2% which is less than 50%. This happened due to the nature of the documents i.e. the contents of almost all of the documents are different news items published by the same publisher, ERTA. Thus, the extent to which these news items will be similar is very rare. Therefore, for most of the queries the number of relevant documents in the refined relevance information set is either 1 or 2. However, for query, "Query 7", 5 relevant documents are returned as the top 5 documents returned by the proposed system and the classical IR systems are taken into consideration. As a result, for other queries that have returns limited number of relevant documents, 1 or 2, the precision goes down because of the false negative 4 or 3 irrelevant documents. This indicates that if the document collection was somehow different, the precision would be much better.

In addition, when we look at the graphs in Figure 5-2, the recall values for both the proposed system and the classical IR goes up and down when we go from the first query to the second query and up to the last. This happened because of the nature of the queries i.e. all the queries are completely different from one another. The recall would have increased linearly if the queries are related to one another i.e. if "Query 2" contains "Query 1" and "Query 3" contains "Query 2" and the same with the rest of the queries.

As it is mentioned in Section 5.3, the harmonic F-measure technique which gives equal weight for recall and precision was used to evaluate both the proposed system and the classical IR system. The harmonic F-measure was chosen over the balanced F-measure because we cannot tell which one of the recall or precision is very important to users. In fact, it is likely that many people may prefer recall to precision. In such case, the balanced F-measure which takes recall as twice as precision is used for evaluation. If this particular balanced F-measure had been used for this study instead of the harmonic F-measure technique, the averaged F-measure would have been much higher than we have now because the recall is 100 %.

Chapter Six - Conclusion and Future work

6.1 Conclusion

Recently, in Ethiopia, huge amount of valuable electronic information in Amharic is being created and published. Some search engines which are dedicated to Amharic language have been developed to retrieve information from repositories. Indexing, one of the components of information retrieval systems, is used to locate documents in an easy and effective way. However, existing indexers for Amharic language have limitations in capturing the semantics of the contents of documents.

In this research, a semantic indexer which is based on domain ontology for Amharic text documents was proposed. The goal of this study is to explore the advantages of ontology and information extraction methods to build a semantic indexer. The concepts and individuals in the document collections are identified using the knowledge stored in the domain ontology. Moreover, new individuals which are not in the ontology are extracted from the document collections using a rule based information extraction technique. All these concepts and individuals are used as index terms to represent documents.

The proposed system is composed of ontology development, document indexing, and query processing components. The ontology development component is intended to come up with football domain ontology. The ontology construction process involves gathering domain terms, designing the concept taxonomy, implementing, evaluating, and populating the ontology. The data used to populate the ontology is collected from EFA. The evaluation processes is made based on expert judgment.

The document indexing process uses the product of the ontology construction process as input. The first major task of the indexing process is tagging documents using concepts and individuals from the ontology. Structured information is extracted from the tagged document collections by using a rule based information extraction technique. This technique applies patterns to build the rules for the information extraction purpose. The patterns are constructed by subject experts

using the tagged document collections. The extracted information contains list of individuals and the relationship among them. Ontology based weighting has been done for all the concepts and individuals found in the documents. The final task in the indexing process is associating the document URLs with the concepts and individuals in the ontology so that the ontology can be used as both a knowledge base and an index. Embedding the index with the ontology helps to minimize the time required to process queries.

The query processing component of the system is developed with the purpose of applying the index in information retrieval (IR) system so that its performance can be evaluated. This component is composed of query tagging, query caching, creating new individuals from query, document retrieval, and ranking activities. Query tagging deals with identifying concepts and individuals from query text. The tagged queries are stored in a query cache so as to avoid the time required to preprocess and tag whenever the same queries are posed. If no single concept or individual is identified by the tagger, those nouns in the query are considered as new individuals and added to the ontology to be used when next time the same query is posed by a user. During document retrieval, inferencing is applied on the concepts and individuals identified by the tagger. The entire original and inferred items are used as index terms and the documents associated with them are retrieved as relevant documents. The retrieved documents are ranked according to their relevance.

The proposed indexer was tested based on the relevance information provided by domain experts. Besides, the proposed system was compared with the classical IR system developed by Tessema [2]. In order to test these two systems, 138 football news articles and 25 queries were used. The precision, recall, and F-measure techniques were used to evaluate the performance of the systems. The proposed semantic indexer has better average recall, precision, and F-measure values compared to the classical IR system.

6.2 Contribution

The contributions of this thesis work are summarized as follows:

- Applying domain ontology to build a semantic indexer, i.e., exploring the advantages of using knowledge base in constructing a semantic document

indexer. Providing a method that embeds ontology and index together so as to minimize query processing time.

- Making use of information extraction technique to capture the embedded information (the semantics) in documents.
- Extending the existing semantic inferencing engine used in the ontology by adding a distance based similarity approach so as to determine the similarity between the original and inferred items (i.e., concepts and individuals). The distance is used to determine the relevance of the inferred items to the query.
- Ontological ranking based on the items' weight and relevance.

6.3 Future Work

In this research, we have made an attempt to explore the use of domain ontology to build a semantic indexer. The following directions are pointed out so that this research can be further pursued.

- As it is mentioned in Section 1.6.2, all words are mapped on to the ontology without applying word sense disambiguation, i.e., without identifying contextual meaning to capture polysemy. Therefore, finding a way to handle polysemy problem should be considered.
- In this study, as stated in Section 4.5, individuals, individual types, siblings, super concepts, and sub concepts are the only items considered for concept reasoning/inferencing. However, applying object properties (relationship between concepts), in addition to the existing items, could be much valuable to enable the system to respond to indirect queries in a better way. For example, referring to the query "የቅዱስ ጊዮርጊስ አሰልጣኝ ስም", the object property "yaseleTnal" that relates the concepts "አሰልጣኝ" and "ቡድን" could be used to catch the semantics of the query, i.e., identifying the individuals of the concept "ቡድን" that is related with the property "yaseleTnal" to the individual "ቅዱስ ጊዮርጊስ" from the ontology the query implies. The documents associated with the individuals (extracted using this object property) are considered as relevant documents. In addition to this, the ontology inferencing algorithm used in this study, gives equal values to the "Hyponym/hypernym" and "Meronym/holonym" relationships.

The algorithm can be further modified to consider those concepts related with Hyponym/hypernym" relationship differently from those related with "Meronym/holonym" relationships.

- As described in Section 4.5, the individuals extracted from queries and not found in the ontology are mapped on to the top concept - "እግር ኳስ". For instance, when we look at the query "የገመና[N] ድራማ[N] አርቲስቶች[N] እና ባለስልጣናት[N]", the terms "የገመና ድራማ አርቲስቶች" and "ባለስልጣናት" are mapped to the concept "እግር ኳስ". However, since these terms are instances of the concept "ቡድን", they should have been mapped to "ቡድን" instead of "እግር ኳስ". Hence, Algorithm 4-9 should be modified to make it capable enough to identify the exact type of the individuals from the ontology.
- We will investigate ways to refine the instance population mechanism; capture new concepts and handle the evolution of existing concepts. In addition, we will investigate on the use of probabilistic approach to rebuild the semantic index and the ontology with minimal effect on the structure of the documents and concepts populated in the semantic index.
- Provide an automatic and generic approach, (rule based or statistical) that extract relevant information from the user query and annotate it with concept from the knowledge base
- The patterns used by Algorithm 4-5 to extract information from documents are built manually. We recommend developing an automated system that automatically generates patterns from documents.

  We believe that the result of this research can be modified with minimal effort to other and similar domain (e.g., agricultural marketing) and hence we will investigate ways to generalize the proposal to other domains.

References

[1] Ralf Steinberger, Johan Hagman, and Stefan Scheer, "Using Thesauri for Automatic Indexing and for the Visualisation of Multilingual Document Collections," in *Proceedings of the Workshop on Ontologies and Lexical Knowledge Bases*, Sozopol, Bulgaria, September, 2000, pp. 130-141.

[2] Tessema Mindaye Mengistu and Solomon Atnafu, "Design and Implementation of Amharic Search Engine," in *International IEEE Conference on Signal-Image Technologies and Internet-Based System - SITIS*, 2009, pp. 318 - 325.

[3] Tewodros Hailemeskel Gebermariam, "Amharic Text Retrieval: An Experiment Using Latent Semantic Indexing (LSI) with Singular Value Decomposition (SVD)," Masters Thesis, School of Information Studies for Africa, Addis Ababa University, Addis Ababa, Ethiopia, Unpublished 2003.

[4] Bethlehem Mengistu Hailemariam, "N-gram-Based Automatic Indexing for Amharic Text," Masters Thesis, Addis Ababa University, Addis Ababa, Unpublished 2002.

[5] S. kara et al., "An Ontology-Based Retrieval System Using Semantic Indexing," *Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on Computing & Processing (Hardware/Software)*, pp. 197-202, 2010.

[6] Parul Gupta and A.Sharma, "Context based Indexing in Search Engines using Ontology," *International Journal of Computer Applications*, Vol. 1, No. 14, pp. 53-56, 2010.

[7] Jacob Kohler, Stephan Philippi, Michael Specht, and Alexander Ruegg, "Ontology based text indexing and querying for the semantic web," *Knowledge Based Systems - KBS*, Vol. 19, No. 8, pp. 744-754, 2006.

[8] Mustapha Baziz, Mohand Boughanem, and Nathalie Aussenac-Gilles, "Evaluating a Conceptual Indexing Method by Utilizing WordNet," *Accessing Multilingual Information Repositories*, pp. 238-246, 2006.

[9] Hassen Redwan, Tessema Mindaye, and Solomon Atnafu, "Enhanced Design of Amharic Search Engine (An Amharic Search Engine with Alias and Multi-character Set Support)," in *AFRICON '09.*, Addis Ababa, 2009, pp. 1-6.

[10] Wolf Leslau, *Amharic Reference Grammar.*: ERIC Clearinghouse for Linguistics, Center for Applied Linguistics,1717 Massachusetts Ave. N.W., Washington, D.C. 20036, 1969.

[11] Tony I. Obaseki, "Automated Indexing: The Key to Information Retrieval in the 21st Century," *Library Philosophy and Practice (e-journal) Libraries at University of Nebraska-Lincoln*, 2010.

[12] Accredited Language Services Blog. [Online]. http://www.alsintl.com/resources/languages/Amharic/ (Accessed on July, 2011)

[13] Meron Sahlemariam, Mulugeta Libsie, and Daniel Yacob, "Concept-Based Automatic Amharic Document Categorization," *AMCIS 2009 Proceedings*, 2009.

[14] Dow Jones Markets, Vijay V. Raghavan, William I. Grosky, Rajesh Kasanagottu, and Venkat N. G Udivada, "Information retrieval on the World Wide Web.," in *IEEE Internet Computing*, 1997.

[15] Gerard Salton, "Syntactic Approaches To Automatic Book Indexing," *Proceedings of the 26th annual meeting on Association for Computational Linguistics. Association for Computational Linguistics*, pp. 204-210, 1988.

[16] S. E. Robertson and K. S. Jones, *Simple, proven approaches to text retrieval*. Cambridge: Computer Laboratory, University of Cambridge, 1997.

[17] Patrick Pantel and Dekang Lin, "A Statistical Corpus-Based Term Extractor," *Advances in Artificial Intelligence*, pp. 36-46, 2001.

[18] Melvin Earl, and John L. Kuhns. Maron, "On Relevance, Probabilistic Indexing and

Information Retrieval," *Journal of the ACM (JACM) 7.3* , Vol. 7, No. 3, pp. 216-244, 1960.

[19] S. E. Robertson, C. J. van Rijsbergen, and M. F. Porter, "Probabilistic models of indexing and searching," *Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, pp. 35-56, 1980.

[20] L Fagan Joel, "Experiments in Automatic Phrase Indexing for Document Retrieval: A comparision of Syntactic and non Syntactic Methods," 1987.

[21] Renee Pohlmann and Wessel Kraaij, "The Effect of Syntactic Phrase Indexing on Retrieval Performance for Dutch Texts," *In Proceedings of RIAO'97*, pp. 176-187, 1997.

[22] Barón Marco Suárez and Valencia Kathleen Salinas, "An approach to semantic indexing and information retrieval," *Revista Facultad de Ingeniería Universidad de Antioquia*, pp. 174-187, 2009.

[23] Barbara Rosario, "Latent Semantic Indexing: An overview," *Techn. rep. INFOSYS 240*, 2000.

[24] Benno Stein, Sven Meyer zu Eissen, and Martin Potthast, "Syntax versus Semantics: Analysis of Enriched Vector Space Models," in *Third International Workshop on Text-Based Information Retrieval (TIR 06)*, University of Trento, Italy, 2006.

[25] Min Chen et al., "Data, Information, and Knowledge in Visualization," *Computer Graphics and Applications, IEEE* , Vol. 29, No. 1, pp. 12-1, 2009.

[26] Randall Davis, Howard Shrobe, and Peter Szolovits, "What Is a Knowledge Representation?," *AI magazine*, Vol. 14, No. 1, p. 17, 1993.

[27] Stephan Grimm, "Knowledge Representation and Ontologies," *Scientific Data Mining and Knowledge Discovery: Principles and Foundations*, 2009.

[28] María Rosario and Bautista Zambrana, "Using Ontologies for the Teaching of Terminology: the case of a package travel Ontology," Spain, 2010.

[29] Nicola Guarino, "Formal Ontology and Information Systems," *Proceedings of FOIS'98, Trento, Italy,. Amsterdam, IOS Press, pp. 3-15.Amsterdam, IOS Press*, pp. 3-15, 6-8 June 1998.

[30] Pretorius A. Johannes, "Ontologies - Introduction and Overview," 2004.

[31] Gunnar O. Klein and Barry Smith, "Concept Systems and Ontologies," *Biomed Inform*, Vol. 39, No. 3, pp. 274-87, 2006.

[32] Natalya F. Noy and Deborah L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*. Californiya: Stanford University, 2001.

[33] Mike Uschold and Michael Gruninger, "Ontologies: Principles, Methods and Applications," *Knowledge Engineering Review*, Vol. 11, No. 2, pp. 93-136, 1996.

[34] R. Gruber Thomas, "Toward Principles for the Design of Ontologies used for Knowledge Sharing," *International Journal of Human Computer Studies*, Vol. 43, No. 5, pp. 907-928, 1995.

[35] Robert van Kralingen, "A Conceptual Frame-based Ontology for the Law," in *Proceedings of the First International Workshop on Legal Ontologies*, 1997.

[36] Oscar Corcho, Mariano Fernandez-Lopez, and Asuncion Gomez-Perez, "Methodologies, tools and languages for building ontologies. Where is their meeting point?," *Data & Knowledge Engineering*, vol. 46, no. 1, pp. 41-64, 2003.

[37] Bechhofer Sean, "Ontology Language Standardization Efforts," *Ontoweb Consortium*, vol. 1, 2002.

[38] Mohammad Mustafa Taye, "Web-Based Ontology Languages and its Based Description Logics," *International Journal of ACM, Jordan*, Vol. 2, No. 1, 2011.

[39] Language & Culture; Amharic Language. [Online]. http://lang.nalrc.wisc.edu/nalrc, (Accessed on March, 2012)

[40] Wimsatt Amanda and Wynn Rachel, "Amharic Language and Culture Manual," Texas, 2011.

[41] M. L., Sydney W. Head, and Roger Cowley Bender, "The Ethiopian Writing," 1976.

[42] Omniglot: The Online Encyclopedia of Writing Systems and Languages. [Online]. http://www.omniglot.com/, (Accessed on February 6, 2013)

[43] Getahun Amare, *ዘመናዊ የአማርኛ ሰዋሰው*. Addis Ababa, 1989.

[44] Baye Yimam, *አጭርና ቀላል የአማርኛ ሰዋሰው*. Addis Ababa: Alpha Printers, 2002.

[45] Mary, Ellen Okurowski, "Information Extraction Overview," *In Proceedings of a Workshop held at Fredericksburg*, pp. 117-121, 1993.

[46] E. Appelt Douglas and J. Israel David, "Introduction to Information Extraction Technology," *AI Communications*, Vol. 12, No. 3, pp. 116-172, 1999.

[47] Michael Gasser, "HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya," in *Conference on Human Language Technology for Development*, Alexandria, Egypt, 2011.

[48] H. Getachew, "The Problems of Amharic Writing System.unpublished," in *A paper presented in advance for the interdisciplinary seminar of the Faculty of Arts and Education. HSIU,* 1967.

[49] Church K Dagan I., "Identifying and translating technical terminology. ," in *In Proceedings of the fourth conference on Applied natural language processing*, 1994, pp. 34-40.

Annex

Annex A: Term Glossary

No	Term	Description
1.	እግር ኳስ Football/Soccer	A game in which two teams of eleven players each contend to get a round ball into the other team's goal primarily by kicking the ball with their feet.
2.	ዳኛ/ Referee	The person of authority who is responsible for presiding over the game from a neutral point of view and making on the fly decisions that enforce the rules of the sport, including sportsmanship decisions.
3.	ዋና ዳኛ/Head referee	The Head Referee is responsible for training, directing and supervising all Referees and Official Scorers. Oversees all scoring processes and procedures. Has final authority for decisions regarding team scores. Play a critical role in ensuring smooth flow of match play, and maintaining the pace of the event.
4.	ረዳት ዳኛ/Assistant referee	The two officials who assist the head referee in making his decisions. They monitor the sidelines and goal-lines to determine when a ball goes out of bounds, when a goal is scored or when players are offside; they use a flag to signal their observations.
5.	የቡድን ባለቤት/Team owner	The one in which the team belongs to or the one who has control over the team.
6.	የቡድን ስራአስኪያኝ/Team manager	The one who plays an extremely important role ensuring the successful management of the team and welfare of the players in his/her care. In the traditional sense, managers are those who make the tactics (long-term and short-term), organize the match-day strategy,

		buys players, scouts players and selects the team on match-day, makes the substitutions, drops players, etc. Basically they are in 'charge' of the team players and staff.
7.	ተጫዋች/Player/Footballer	A football player or footballer is a sportsperson who plays in a particular football.
8.	ተባባሪዎች/Sponsors	A person or organization that provides funds or support for the team.
9.	የህክምና ባለሞያ/ ወጌሻ/Team Physician	The one who has the professional ability to provide support when the players get injured on the field.
10.	አምበል/Captain	A title given to a member of the team and who has significant responsibility for strategy and teamwork while the game is in progress on the field.
11.	አህጉር/Continent	Large, continuous, discrete masses of land, ideally separated by expanses of water.
12.	ሀገር /Country	A nation with its own government, occupying a particular territory.
13.	ምድብ/Group	A group contains number of teams from a certain competition. During a cup tournament, teams are organized in groups. Teams in one group play with each other so as to win and pass to the next round.
14.	እግር ኳስ ፌዴሬሽን/Football federation	A governing body of association football.
15.	የ አህጉር እግር ኳስ ፌዴሬሽን/Continent's football federation	A governing body of association football for different regions of a particular continent.
16.	የ ሀገር እግር ኳስ ፌዴሬሽን/ Country's football federation	A governing body of association football for different regions of a particular country.
17.	የ አለም እግር ኳስ ፌዴሬሽን/ world's football federation	A governing body of association football for different regions and continents throughout the world.
18.	አሰልጣኝ/Coach	A person who trains the team in fitness, skills, strength and conditioning, etc.

19.	ዋና አሰልጣኝ/Head Coach	The member of the coaching staff that is responsible for all aspects of the team, and is in charge of all other coaches.
20.	ረዳት አሰልጣኝ/Assistant Coach	The coaches that specialize in specific areas of the team and are directly under the supervision of the head coach.
21.	ቡድን/Team	The collective name given to a group of players selected together to form one side in a football competition.
22.	ብሄራዊ ቡድን/National Team	An all-star team that represents a country in the various international tournaments -- e.g., the World Cup, the Olympic Games, the under-20 World Cup, etc. National teams are supposed to consist of the very best players in the country, regardless of which club they play for. They are not permanent teams; they are assembled only to play in specific games or tournaments.
23.	የተጫዋች ኩኔታ/Player's event	An event which is performed by players during a match like scoring a goal, saving a goal, etc.
24.	ጨዋታ/ግጥሚያ/Match	A formal sports event in which teams compete to win.
25.	ሽልማት/ስጦታ/Reward	An award given to the players or the team due to different circumstances.
26.	የዳኛ ኩኔታ/Referee's event	An event performed by referee's during a match like starting a game, booking, etc
27.	ዘመን/Season	A period of the year where football is played.
28.	ቆይታ/Duration	The time spent from the starting of the match up to the end.
29.	ውድድር/Competition/Tournament	A series of sport events in which a winner is selected from among the two entrants.
30.	ሊግ/League	A group of teams or players who regularly compete against one another which are put in order according to how

		many points they have won.
31.	የ ዋንጫ ጨዋታ/Cup	The final game where a cup will be awarded to the winner of the game.
32.	ቦታ/Place	A term used to refer to city, country town or position.
33.	የተጫዋች ቦታ/Player position	The place where the players reside.
34.	ግብ ጠባቂ/በረኛ/Goal kepper	A designated player charged with directly preventing the opposing team from scoring by intercepting shots at goal.
35.	የመሀል ተጫዋች / Midfielder	The two, three or four players who link together the offensive and defensive functions of a team. Midfielders play in front of the defenders and behind the forwards.
36.	የቀኝ ተመላላሽ/Right wing	The outside forwards who play close to the sidelines whose primary task is to provide the strikers with accurate crossing passes so they can shoot at the goal.
37.	የግራ ተመላላሽ/Left wing	The outside forwards who play close to the sidelines whose primary task is to provide the strikers with accurate crossing passes so they can shoot at the goal.
38.	ተከላካይ / Defender(Defense)	A player who functions primarily in the defensive third of the field and whose major role is to fend off attacks on the goal by the opposing team.
39.	የግራ ተከላካይ/Left back	A defender who plays primarily in a position on the left side of the field
40.	የቀኝ ተከላከይ/Right back	A defender who plays primarily in a position on the right side of the field
41.	አጥቂ/offense/forward/striker	A player on a team who plays nearest to the opposing team's goal, and is therefore principally responsible for scoring goals.
42.	የመሀል አጥቂ/Center forward	An offensive player who covers the center of the field and who usually starts the kickoff.

43.	የቀኝ አጥቂ/Right forward	The player with the ball located in the right side of the field and who is trying to score.
44.	የግራ አጥቂ/Left forward	The player with the ball located in the left side of the field and who is trying to score.
45.	ኳስ ሜዳ/የ እግር ኳስ መጫወቻ ሜዳ/የስፖርት ማዘዉተርያ ሜዳ/Stadium	A large, usually open structure for sports events with tiered seating for spectators.
46.	የተመልካች መቀመጫ/Audiences' seat	The place where the audience of the game seats in.
47.	ሜዳ/Field	The playing area on which football is played
48.	የመሀል መስመር/Center line	The line in the center of the field where the game gets started.
49.	የጎን መስመር/Side line	The line in the left and right side of the field where the assistant referees reside.
50.	የመጨረሻ መስመር/End line	A line at either end of the field 10 yards beyond and parallel to the goal line.
51.	ማእዘን/Corner	The position at which two lines, surfaces, or edges meet and form an angle in the field.
52.	የፍጹም ቅጣት ምት መምቻ ክልል/Penalty area	The area where a penalty will be given if a foul is committed.
53.	የጎል ክልል/Goal area	The area where the probability of a goal being scored is high.
54.	መሀል ሜዳ/Center	The center of the field where the game is started.
55.	የወዳጅነት ጨዋታ/የአቋም መለኪያ ጨዋታ	A match where teams play in order to assess their level of competency so as to get ready for the real match.
56.	የቡድን አመራር / Team management	The way the team is guided through each and every circumstance and how they handle difficulties whenever they face it.
57.	የበረኛ አሰልጣኝ/ The goal keeper's coach	The coach who guides and trains the goal keeper so as to do his/her job properly.
58.	ትጥቅ/Kit/Football equipment	All the necessary equipment needed for each player to play, like their jersey, shoes...
59.	ኳስ ወደ ዉጭ ማዉጣት/Kick out	Hitting the ball in order to take it outside of the field.

60.	ወደ ዉጭ የወጣ ኳስ መወርወር/የእጅ ውርወራ/Throw in	When the player throws the ball using his/her hand because the ball has been outside of the field.
61.	ኳስ ማንከባለል/Move ball	Moving the ball slowly to keep it in possession.
62.	ኳስ መቀበል/Get ball	Apprehending the ball from another teammate.
63.	ኳስ መቆጣጠር /Controll ball	Keeping the ball under control so as the opposite player won't take it away.
64.	ኳስ በእጅ መንካት/ Handball	A foul where a player touches the ball with his hand or arm; depending on where the offence take place, the opposing team is awarded either a penalty kick or a direct free kick.
65.	የፍጹም ቅጣት ምት/Penalty kick	A kick taken from the penalty spot by a player against the opposing goalie. Awarded for the most severe rule violations and those committed by defenders within their own penalty area.
66.	የማእዘን ምት /Corner kick	A restart of the game where the ball is kicked from the corner arc into the middle of the penalty area in an attempt to create a scoring chance. Awarded to an attacking team when the ball crosses the defending team's goal-line after being last touched by the defending team.
67.	ኳስ በራስ መግጨት/Overhead kick	An event where the player hits the ball with his/her forehead.
68.	ኳስ በደረት ማቆም/Chest trap	An event where the player controls the ball with his/her chest.
69.	መሮጥ/Running	An event where the player runs in order to have the ball in possession.
70.	ቀስ ብሎ መሄድ/walking slowly	Moving slowly in order to waste the time or to calm their selves up.
71.	ኳስ መያዝ/ Own ball	Having the ball in possession.
72.	ኳስ ማንጠር/Dribble	Moving the ball up and down while getting ready to throw it or to hit it.
73.	ኳስ መለጋት/kicking ball	Hitting the ball in possession to give it to another player.

74.	ኳስ በእጅ ወደ ውጭ ማውጣት/Ball out	Touching the ball using their hand in order to make the ball outside of the field.
75.	መውደቅ/Falling down	Falling down in the field because a foul has been made from the opposite team.
76.	ኦፍሳይድ/Offside	The situation where an attacking player, on the offensive half of the field, has put himself in a position where there are fewer than two opponents (usually the goalie and one defender) between him and the goal at the exact moment the ball is kicked forward. This positioning does not constitute a foul until he becomes involved in the play. A player is not offside if he is exactly even with one or both of these defensive players.
77.	መልሶ ማጥቃት / Counter attack	An attack launched by a defending team immediately after it regains possession of the ball. A counter attack in soccer is equivalent to a fast break in basketball.
78.	ጨዋታ ማስጀመር /Starting the game	When the referee blows the whistle to indicate the match has started.
79.	ጨዋታ ማስፈጸም/Ending the game	When the referee blows the whistle to indicate the match has ended.
80.	ማስጠንቀቂያ/Warning	When the referee gives a warning to a player due to a foul he/she made, not to repeat it next time.
81.	የባከነ ሰአት መጨመር/adding extra time	When the referee adds an additional time at the end of a match, to compensate for time lost through injury or other circumstances.
82.	ቀይ ካርድ/Red card	A red card that a referee holds up to signal a player's expulsion from the game; the player's team must then play the rest of the game shorthanded. Presented for violent behavior or multiple infractions (two yellow cards = one red card).
83.	ቢጫ ካርድ/Yellow card	A yellow card that a referee holds up to warn a player for dangerous or un-sportsmanlike behavior; also known as a caution. Two yellow cards in one game

		earn a player an automatic red card, signaling his expulsion.
	ጨዋታ ማቋረጥ/ Withdrawing a game	When a player withdraws a game while playing due to injury or some other reason
84.	ፍፃሜ/የዋንጫ ጨዋታ/Final	A match taking place immediately after the semifinal.
	ሩብ ፍፃሜ/Quarter final	One of four competitions in a tournament, whose winners go on to play in semifinal competitions.
	ግማሽ ፍፃሜ /Semi final	A match taking place immediately before the final.
85.	የመጀመሪያ አጋማሽ/የመጀመርያ 45 ደቂቃ/First half	The first 45 minutes of two halves of play.
86.	የሁለተኛ አጋማሽ/ የሁለተኛ 45 ደቂቃ/ Second half	The second 45 minutes of two halves of play.
87.	ተጨማሪ ሰአት/የባከነ ሰዓት//Extra time	An additional period played at the end of a match, to compensate for time lost through injury or (in certain circumstances) to allow the teams to achieve a conclusive result.
88.	ነጥብ/Points	Awarded for results attained from matches. Three points are commonly awarded for a victory, one for a draw and none for a defeat. These points determine a team's positioning in a league.
89.	ጥፋት/Foual	A violation of the rules - including kicking, pushing, shoving, tripping and dangerous or aggressive play - for which an official awards a free kick.
90.	ዋንጫ/Trophy	A prize in sports such as a cup or plaque, received as a symbol of victory.
91.	የተቀያሪ መቀመጫ / Substitutes' bench	Generally occupied by a team's non-playing members of staff during a game. The manager, assistant managers, coaches, physicians and substitutes commonly sit on the bench during a match. An actual bench is not as widely used these days, with clubs introducing comfy seats in the dugout.
92.	በበረኛ የሚመታ የመልስ ምት/Goal kick	A type of restart in which the ball is

		kicked from inside the goal area away from the goal. Awarded to the defending team when a ball that crossed its goal-line was last touched by a player on the attacking team.
93.	የተጫዋች ለዉጥ/substitution	Replacement of one player on the field with another player not on the field. Teams are allowed three substitutions per game.

Annex B: List of Concepts

No	Concept
1.	FootBall:Igr_kWas
2.	FootBall:Igr kWas fEdErExn
3.	FootBall:aTqi
4.	FootBall:ahgur
5.	FootBall:ambel
6.	FootBall:aselTaN
7.	FootBall:Cewata
8.	FootBall:balebEt
9.	FootBall:bhErawi_budn
10.	FootBall:bota
11.	FootBall:budn
12.	FootBall:daNa
13.	FootBall:Degafi
14.	FootBall:gb_Tebaqi
15.	FootBall:hager
16.	FootBall:kWas_mEda
17.	FootBall:kewaN
18.	FootBall:kunEta
19.	FootBall:lig
20.	FootBall:mEda
21.	FootBall:maIzen
22.	FootBall:Mdb
23.	FootBall:mehal mEda
24.	FootBall:Mereb
25.	FootBall:Qoyta
26.	FootBall:redat_aselTaN
27.	FootBall:redat_daNa
28.	FootBall:sraaskiyaj
29.	FootBall:teCawac
30.	FootBall:tebabariwoc
31.	FootBall:tekelakay
32.	FootBall:wanCa
33.	FootBall:wana_aselTaN

34.	FootBall:wana_daNa
35.	FootBall:wddr
36.	FootBall:wedajnet
37.	FootBall:xlmat
38.	FootBall:ye_ahgur_Igr_kWas_ fEdErExn
39.	FootBall:ye alem Igr kWas fEdErExn
40.	FootBall:ye hager Igr kWas fEdErExn
41.	FootBall:ye lig zur
42.	FootBall:yebereNa aselTaN
43.	FootBall:yebudn amerar
44.	FootBall:yedaNa_kunEta
45.	FootBall:yefSum qTat mt memca kll
46.	FootBall:yegol kll
47.	FootBall:yegon Mesmer
48.	FootBall:yegra aTqi
49.	FootBall:yegra tekelakay
50.	FootBall:yegra temelalax
51.	FootBall:yehkmna balemuya
52.	FootBall:yemeCerexa Mesmer
53.	FootBall:yemehal_aTqi
54.	FootBall:yemehal Mesmer
55.	FootBall:yemehal_teCawac
56.	FootBall:yeqeN_aTqi
57.	FootBall:yeqeN_tekelakay
58.	FootBall:yeqeN_temelalax
59.	FootBall:yeteCawac_bota
60.	FootBall:yeteCawac_kunEta
61.	FootBall:yetemelkac meqemeCa
62.	FootBall:yewanCa_zur
63.	FootBall:zemen
64.	FootBall:zur

Annex C: List of Object Properties

No	Object Properties
1.	FootBall:alewu_balebEt
2.	FootBall:alew_teCawac
3.	FootBall:alew_tebabari
4.	FootBall:alew_wddr
5.	FootBall:balebEt_newu_ye
6.	FootBall:be_amet
7.	FootBall:be_zur
8.	FootBall:has_Rank_and_Point
9.	FootBall:has_leader
10.	FootBall:has_link
11.	FootBall:has_weight
12.	FootBall:in_year
13.	FootBall:sra_askiyaj_newu_ye

14.	FootBall:wana_kstetoce_be
15.	FootBall:yCawetal_le
16.	FootBall:yaseleTnal
17.	FootBall:yastenagdal
18.	FootBall:ydaNal
19.	FootBall:yetekahEdebet_se'at
20.	FootBall:yeteseTebet_amet
21.	FootBall:yeteseTebet_seat
22.	FootBall:ygeNal_be
23.	FootBall:ygebal_ye
24.	FootBall:ykahEdal_be
25.	FootBall:ykenawenal_be
26.	FootBall:ykesetal_be
27.	FootBall:ymeral_be
28.	FootBall:ynorewal_balebEt
29.	FootBall:ynorewal_gTmiya
30.	FootBall:ynorewal_teCawac
31.	FootBall:ynorewal_yewanCa_zur
32.	FootBall:ynorewal_zur
33.	FootBall:ysatefal
34.	FootBall:yseTal
35.	FootBall:yseleTnal_be
36.	FootBall:yweklal_ahugrn
37.	FootBall:yweklal_klebocn_ke

Annex D: List of Data Type Properties

No	Data Type Properties
1.	FootBall:yedaNa_adraxa
2.	FootBall:demoz
3.	FootBall:kokeb_gb_agbi
4.	FootBall:meri_budn_na_wuTEt
5.	FootBall:qSl_sm
6.	FootBall:wuTEt
7.	FootBall:yeCaweta_mejemerya_seat
8.	FootBall:yekstet_aynet
9.	FootBall:yetwuld_bota
10.	FootBall:yetwuld_qen
11.	FootBall:yexlmat_aynet
12.	FootBall:yettq_qutr
13.	FootBall:zEgnet

Annex E: Patterns

Category Number	Category Name	Pattern
1	Match-Result	<የሊግ ዙር> +< ሀገር> + <ሊግ> +<ቡድን>+<ቡድን>ን+ <ውጤት>+ "አሸንፎአል/አሸነፈ"
		<የሊግ ዙር> + < ሀገር> + <ሊግ>+ <ቡድን>+ በ<ቡድን> + <ዉጤት> "ተሸንፏል"
		<የሊግ ዙር> + < ሀገር> + <ሊግ>+ <ቡድን> + ከ<ቡድን> + <ዉጤት> " ተለያይተዋል"
		<የሊግ ዙር> + <ሀገር> + <ሊግ>+ <ቡድን>ን + ያስተናገደው+ <ቡድን>+ <ዉጤት> "አሸንፎአል"
		<የሊግ ዙር> + < ሀገር> + <ሊግ>+ <ቡድን>ን + ያስተናገደው+ ቡድን>+ <ዉጤት> "ተሸንፏል"
		<ካስ ሜዳ> የተገናኙት <ቡድን>ና + <Team> + <ዉጤት>ተለያይተዋል
		<ቡድን> + <የሊግ ዙር> + <NoOfGames> + <ውጤት>ተሸንፏል
		<ቡድን> + <የዋንጫ ዙር> + <NoOfGames> + <ውጤት>ተሸንፏል
		<ቡድን> + <የሊግ ዙር> + <NoOfGames> +<ዉጤት>አሸንፎአል
		<ቡድን> + <የዋንጫ ዙር> + <NoOfGames> + <ውጤት>ተሸንፏል
		<የሊግ ዙር> + < ሀገር> + <ሊግ>+ <ቡድን>ን + በ<ካስ ሜዳ> ያስተናገደው+ <ቡድን>+ <ውጤት>"ተሸንፏል"
		<የሊግ ዙር> + < ሀገር> + <ሊግ>+ <ቡድን>ን + በ<ካስ ሜዳ> ያስተናገደው+ <ቡድን>+ <ውጤት>"አሸንፎአል"
		የ<ሀገር> + <ቡድን>+ የ<ሀገር> + <ቡድን>+ በ<ካስ ሜዳ> አስተናግዶ + <ውጤት> ተለያይቷል
		<ሀገር> + <ሊግ> + የ<ሀገር> + <ቡድን>+ የ<ሀገር> + <ቡድን>+ በ<ካስ ሜዳ> + አስተናግዶ + <ውጤት> ተሸንፏል
		<ሀገር> + <ሊግ>+ <ሀገር> + <ቡድን>+ <ሀገር> + <ቡድን>+ በ<ካስ ሜዳ> + አስተናግዶ + <ውጤት> አሸንፏል
		<ሀገር> + <ሊግ>+ <የጨዋታ መጀመሪያ ሰዓት> + <ቡድን>+ ከ<ቡድን>በ <ካስ ሜዳ> + ተገናኝተዋል
		< የሊግ ዙር> + <ሊግ>+ <ቡድን>+ <ቡድን> + <NoOfGames> + <ውጤት>+ ተለያይተዋል
		<ሀገር> + <ሊግ> <የሊግ ዙር> + <ከተማ> + <ቡድን> + <ቡድን>ን + <ውጤት> + አሸንፏል
		<ሀገር> + <ሊግ> <የሊግ ዙር> + <ከተማ> + <ቡድን> + በ<ቡድን> + <ውጤት> + ተሸንፏል
		<ሀገር> + <ሊግ> <የሊግ ዙር> + <ከተማ> + <ቡድን> + ከ<ቡድን> + <ውጤት> + ተለያይተዋል
		<የሊግ ዙር> + <ሊግ> + <ሀገር> + <ብሄራዊ ቡድን> + <ሀገር>+ <ብሄራዊ ቡድን>ን + <ውጤት>+ አሸንፏል/አሸንፎአል/አሸነፈ
		<የሊግ ዙር> + <ሊግ> + <ሀገር> + <ብሄራዊ ቡድን> + <ሀገር>+ በ<ብሄራዊ ቡድን> + <ውጤት>+ ተሸንፏል
		<የሊግ ዙር> + <ሊግ> + <ሀገር> + <ብሄራዊ ቡድን> + <ሀገር>+ <ብሄራዊ ቡድን> + < ውጤት> + ተለያይተዋል
		<የሊግ ዙር> +ከm<ሊግ> <ሀገር> <ብሄራዊ ቡድን> + <ሀገር>+ <ብሄራዊ ቡድን> + <ካስ ሜዳ> + ተገናኝተው + < ውጤት> + ተለያይተዋል

		<የሊግ ዙር> <ሊግ> <ሀገር> <ብሄራዊ ቡድን>ን + “ያተናገደው” +<ሀገር>+ <ብሄራዊ ቡድን> + <ውጤት>+ አሸንፏል/አሸንፎአል/አሸነፈ
		<የሊግ ዙር> <ሊግ> <ሀገር> <ብሄራዊ ቡድን>ን + “ያተናገደው” +<ሀገር>+ <ብሄራዊ ቡድን> + ውጤት+ ተሸንፏል
		የ<ሀገር> + <ብሄራዊ ቡድን>+ የ<ሀገር> + <ብሄራዊ ቡድን>+ በ<ካስ ሜዳ> አስተናግዶ + <ውጤት> ተለያይቷል
		< የሊግ ዙር> + <ሊግ>+ <ሀገር> + <ብሄራዊ ቡድን> + <ሀገር> + <ብሄራዊ ቡድን> + በ<ካስ ሜዳ> + አስተናግዶ + <ውጤት> አሸንፏል
		< የሊግ ዙር> + <ሊግ>+ <ሀገር> + <ብሄራዊ ቡድን> + ከ<ሀገር> +<ብሄራዊ ቡድን> + < የጨዋታ መጀመሪያ ሰዓት> + በ<ካስ ሜዳ> + ተገናኝተዋል
		< የሊግ ዙር> + <ሊግ>+ <ሀገር> + <ብሄራዊ ቡድን> + <ሀገር> + <ብሄራዊ ቡድን> <NoOfGames> <ውጤት>+ ተለያይተዋል
		< የሊግ ዙር> + <ሊግ> + <ሀገር> + <ብሄራዊ ቡድን> + <ሀገር> + <ብሄራዊ ቡድን>ን + <ከተማ> + <ውጤት> + አሸንፏል
		< የሊግ ዙር> + <ሊግ> + <ሀገር> + <ብሄራዊ ቡድን> + በ<ሀገር> + <ብሄራዊ ቡድን> + <ከተማ> + <ውጤት> + ተሸንፏል
		< የሊግ ዙር> + <ሊግ> + <ሀገር> + <ብሄራዊ ቡድን> + ከ<ሀገር> + <ብሄራዊ ቡድን> + <ከተማ> + <ውጤት> ተለያይተዋል
		<ዘመን> + <ዋንጫ> + በ <ሀገር> ይካሄዳል
		<የዋንጫ ዙር> +< ሀገር> + <ዋንጫ> +<ቡድን>+<ቡድን>ን+ <ውጤት>+ "አሸንፎአል/አሸነፈ"
		<የዋንጫ ዙር> + < ሀገር> + <ዋንጫ>+ <ቡድን>+ በ<ቡድን>+ <ውጤት>"ተሸንፏል"
		<የዋንጫ ዙር> + < ሀገር> + <ዋንጫ>+ <ቡድን>+ ከ<ቡድን>+ <ውጤት>" ተለያይተዋል"
		<የዋንጫ ዙር> + < ሀገር> + <ዋንጫ>+ <ቡድን>ን + ያስተናገደው+ <ቡድን>+ <ውጤት> "አሸንፎአል"
		<የዋንጫ ዙር> + < ሀገር> + <ዋንጫ>+ <ቡድን>ን + ያስተናገደው+ <ቡድን>+ ውጤት"ተሸንፏል"
		<የዋንጫ ዙር> + < ሀገር> + <ዋንጫ>+ <ቡድን>ን + በ<ካስ ሜዳ> ያስተናገደው+ <ቡድን>+ <ውጤት> "ተሸንፏል"
		<የዋንጫ ዙር> + < ሀገር> + <ዋንጫ>+ <ቡድን>ን + በ<ካስ ሜዳ> ያስተናገደው+ <ቡድን>+ <ውጤት> "አሸንፎአል"
		<ሀገር> + <ዋንጫ> + የ<ሀገር> + <ቡድን>+ የ<ሀገር> + <ቡድን>+ በ<ካስ ሜዳ> + አስተናግዶ + <ውጤት> ተሸንፏል
		< የዋንጫ ዙር> + <ዋንጫ>+ <ቡድን>+ <ቡድን> <NoOfGames> ውጤት + ተለያይተዋል
		<ሀገር> <ዋንጫ> + <የዋንጫ ዙር> + <ከተማ> + <ቡድን> + <ቡድን>ን + <ውጤት> + አሸንፏል
		<ሀገር> <ዋንጫ> + <የዋንጫ ዙር> + <ከተማ> + <ቡድን> + በ<ቡድን> + <ውጤት> + ተሸንፏል
		<ሀገር> <ዋንጫ> + <የዋንጫ ዙር> + <ከተማ> + <ቡድን> + ከ<ቡድን> + <ውጤት> + ተለያይተዋል
		<የዋንጫ ዙር> + <ዋንጫ> <ሀገር> + <ብሄራዊ ቡድን> + <ሀገር>+ <ብሄራዊ ቡድን>ን + <ውጤት>+ አሸንፏል/አሸንፎአል/አሸነፈ
		<የዋንጫ ዙር> + <ዋንጫ> <ሀገር> <ብሄራዊ ቡድን> + <ሀገር>+ በ<ብሄራዊ ቡድን> + <ውጤት>+ ተሸንፏል
		<የዋንጫ ዙር> + < ዋንጫ> + <ሀገር> + <ብሄራዊ ቡድን> + <ሀገር>+ <ብሄራዊ ቡድን> + < ውጤት> + ተለያይተዋል
		<የዋንጫ ዙር> + < ዋንጫ> <ሀገር> <ብሄራዊ ቡድን> + <ሀገር>+ <ብሄራዊ ቡድን> + <ካስ ሜዳ> + ተገናኝተው + < ውጤት> + ተለያይተዋል
		<የዋንጫ ዙር> + < ዋንጫ> + < ሀገር> + <ብሄራዊ ቡድን>ን + “ያተናገደው” +<ሀገር>+ <ብሄራዊ ቡድን> + ውጤት+ አሸንፏል/አሸንፎአል/አሸነፈ

		<የዋንጫ ዙር> < ዋንጫ> <ሀገር> <ብሄራዊ Team>ን + “ያተናገደው” +<ሀገር>+ <ብሄራዊ ቡድን> + ውጤት+ ተሸንፏል
		<የዋንጫ ዙር> < ዋንጫ> የ<ሀገር> + <ቡድን>+ የ<ሀገር> + <ብሄራዊ ቡድን>+ በ<ኳስ ሜዳ> አስተናግዶ + <ውጤት> ተለያይቷል
		< የዋንጫ ዙር> + < ዋንጫ>+ <ሀገር> + <ብሄራዊ ቡድን> + <ሀገር> + <ብሄራዊ ቡድን> + በ<ኳስ ሜዳ> + አስተናግዶ + <ውጤት> አሸንፏል
		< የዋንጫ ዙር> + < ዋንጫ>+ <ሀገር> + <ብሄራዊ ቡድን> + ከ<ሀገር> +<ብሄራዊ ቡድን> + < የጨዋታ መጀመሪያ ሰዓት> + በ<ኳስ ሜዳ> + ተገናኝተዋል
		< የዋንጫ ዙር> + < ዋንጫ>+ <ሀገር> + <ብሄራዊ ቡድን> + <ሀገር> + <ብሄራዊ ቡድን> <NoOfGames> <ውጤት>+ ተለያይተዋል
		< የዋንጫ ዙር> + < ዋንጫ> + <ሀገር> + <ብሄራዊ ቡድን> + <ሀገር> + <ብሄራዊ ቡድን>ን + <ከተማ> + <ውጤት> + አሸንፏል
		< የዋንጫ ዙር> + < ዋንጫ> + <ሀገር> + <ብሄራዊ ቡድን> + በ<ሀገር> + <ብሄራዊ ቡድን> + <ከተማ> + <ውጤት> + ተሸንፏል
		< የዋንጫ ዙር> + < ዋንጫ> + <ሀገር> + <ብሄራዊ ቡድን> + ከ<ሀገር> + <ብሄራዊ ቡድን> + <ከተማ> + <ውጤት> ተለያይተዋል
2	Match-Player-Event	<የሊግ ዙር> +< ሀገር> + <ሊግ> + <ሀገር> +<ኳስ ሜዳ> +<ቡድን> +<ቡድን>+ <ቆይታ> + <ደቂቃ> + <ተጫዋች>+ ለ<ቡድን> + <ጎል>
		<የሊግ ዙር> +< ሀገር> + <ሊግ> +<ሀገር> +<ኳስ ሜዳ>+<ቡድን> +<ቡድን>+ <ቆይታ> + <ደቂቃ> + የ<ቡድን> + <ተጫዋች>+ <ጎል>
		<የዋንጫ ዙር> +< ሀገር> + <ዋንጫ> +<ከተማ>+<ኳስ ሜዳ> +<ቡድን> +<ቡድን>+ <ቆይታ> + <ደቂቃ> + <ተጫዋች>+ ለ<ቡድን> + <ጎል>
		<የዋንጫ ዙር> +< ሀገር> + <ዋንጫ> +<ከተማ> +<ኳስ ሜዳ>+<ቡድን> +<ቡድን>+ <ቆይታ> + <ደቂቃ> + የ<Team> + <ተጫዋች>+ <ጎል>
		<የሊግ ዙር> +< ሀገር> + <ሊግ>+<ከተማ>+<ኳስ ሜዳ> +<ቡድን> +<ቡድን>+ <ቆይታ> + <ደቂቃ> + የ<ቡድን> + <ተጫዋች+ <ኳስ ማሳለፍ> + <ተጫዋች> + <ጎል>
		<የሊግ ዙር> +< ሀገር> + <ሊግ>+<ሀገር>+<ኳስ ሜዳ> +<ቡድን> +<ቡድን>+ <ቆይታ> + <ደቂቃ> + <ተጫዋች+ <ኳስ ማሳለፍ> + <ተጫዋች> + <ቡድን> + ለ<ጎል>
		<የዋንጫ ዙር> +< ሀገር> + <ዋንጫ> +<ከተማ>+<ኳስ ሜዳ> +<ቡድን> +<ቡድን>+ <ቆይታ> + <ደቂቃ> + የ<ቡድን> + <ተጫዋች+ <ኳስ ማሳለፍ> + <ተጫዋች> + <ጎል>
		<የዋንጫ ዙር> +< ሀገር> + <ዋንጫ>+<ከተማ> +<ኳስ ሜዳ>+<ቡድን> +<ቡድን>+ <ቆይታ> + <ደቂቃ> + <ተጫዋች+ <ኳስ ማሳለፍ> + <ተጫዋች> + <ቡድን> + ለ<ጎል>
		<የሊግ ዙር>+ <ሊግ> +<ከተማ> +<ኳስ ሜዳ>+< ሀገር> + <ብሄራዊ ቡድን> +< ሀገር> + < ብሄራዊ ቡድን>+ <ቆይታ> + <ደቂቃ> + <ተጫዋች>+ ለ< ብሄራዊ ቡድን> + <ጎል>
		<የሊግ ዙር>+ <ሊግ> +<ከተማ>+<ኳስ ሜዳ> +< ሀገር> + <ብሄራዊ ቡድን> +< ሀገር> + < ብሄራዊ ቡድን>+<ቆይታ> + <ደቂቃ> + የ< ብሄራዊ ቡድን> + <ተጫዋች>+ <ጎል>
		<የዋንጫ ዙር>+ <ዋንጫ> +<ከተማ> +<ኳስ ሜዳ>+< ሀገር> + <ብሄራዊ ቡድን> +< ሀገር> + < ብሄራዊ ቡድን>+<ቆይታ> + <ደቂቃ> + <ተጫዋች>+ ለ< ብሄራዊ ቡድን> + <ጎል>
		<የዋንጫ ዙር>+ <ዋንጫ> +<ከተማ>+<ኳስ ሜዳ> +< ሀገር> + <ብሄራዊ ቡድን> +< ሀገር> + < ብሄራዊ ቡድን>+<ቆይታ> + <ደቂቃ> + የ< ብሄራዊ ቡድን> + <ተጫዋች>+ <ጎል>
		<የሊግ ዙር>+ <ሊግ> +<ከተማ>+<ኳስ ሜዳ> +< ሀገር> + <ብሄራዊ ቡድን> +< ሀገር> + < ብሄራዊ ቡድን>+<ቆይታ> + <ደቂቃ> + የ< ብሄራዊ ቡድን> + <ተጫዋች+ <ኳስ ማሳለፍ> + <ተጫዋች> + <ጎል>
		<የሊግ ዙር>+ <ሊግ>+<ከተማ>+<ኳስ ሜዳ> +< ሀገር> + <ብሄራዊ ቡድን> +< ሀገር> + < ብሄራዊ ቡድን>+<ቆይታ> + <ደቂቃ> + <ተጫዋች+ <ኳስ ማሳለፍ> + <ተጫዋች> + < ብሄራዊ ቡድን> + ለ<ጎል>
		<የዋንጫ ዙር>+ <ዋንጫ> +<ከተማ> +<ኳስ ሜዳ>+< ሀገር> + <ብሄራዊ ቡድን> +< ሀገር> + < ብሄራዊ ቡድን>+<ቆይታ> + <ደቂቃ> + የ< ብሄራዊ ቡድን> + <ተጫዋች+ <ኳስ ማሳለፍ> +

		<ተጫዋች> + <ጎል>
		<የዋንጫ ዙር>+ <ዋንጫ>+<ከተማ>+<ኳስ ሜዳ> +< ሀገር> + <ብሄራዊ ቡድን> +< ሀገር> + < ብሄራዊ ቡድን>+<ቆይታ> + <ደቂቃ> + <ተጫዋች+ <ኳስ ማሳለፍ> + <ተጫዋች> + < ብሄራዊ ቡድን> + ለ<ጎል>
3	Match-Referee-Event	<የሊግ ዙር>+ <ሊግ>+<ከተማ>+<ኳስ ሜዳ> +< ሀገር> + <ቡድን> +< ሀገር> + <ቡድን>+<ቆይታ> + <ደቂቃ> + <ዳኛ> + <ተጫዋች+ <የዳኛ ኩኔታ>
		<የዋንጫ ዙር>+ <ዋንጫ>+<ከተማ>+<ኳስ ሜዳ> +< ሀገር> + <ቡድን> +< ሀገር> + <ቡድን>+<ቆይታ> + <ደቂቃ> + <ዳኛ> + <ተጫዋች+ <የዳኛ ኩኔታ>
		<የሊግ ዙር>+ <ሊግ>+<ከተማ>+<ኳስ ሜዳ> +< ሀገር> + <ብሄራዊ ቡድን> +< ሀገር> + < ብሄራዊ ቡድን>+<ቆይታ> + <ደቂቃ> + <ዳኛ> + <ተጫዋች+ <የዳኛ ኩኔታ>
		<የዋንጫ ዙር>+ <ዋንጫ>+<ከተማ>+<ኳስ ሜዳ> +< ሀገር> + <ብሄራዊ ቡድን> +< ሀገር> + < ብሄራዊ ቡድን>+<ቆይታ> + <ደቂቃ> + <ዳኛ> + <ተጫዋች+ <የዳኛ ኩኔታ>
4	Competition-Team-Rank	<ዘመን> + <የሊግ ዙር> + <ሀገር> + <ሊግ> + <ቡድን> + <ደረጃ>
		<ዘመን> + <የሊግ ዙር> + <ሀገር><ሊግ><ብሄራዊ ቡድን>+ <ደረጃ>
		<ዘመን> + <የዋንጫ ዙር> + <ሀገር> + <ዋንጫ> + <ቡድን>+ <ደረጃ>
		<ዘመን> + <የዋንጫ ዙር> + <ሀገር> + <ዋንጫ> + <ብሄራዊ ቡድን> + <ደረጃ>
5	Competition-Player-Point	<ዘመን> + <የሊግ ዙር> + <ሀገር> + <ሊግ> “ኮከብ ግብ አግቢ” <ቡድን><ተጫዋች> + <ነጥብ>
		<ዘመን> + <የሊግ ዙር> + <ሀገር> + <ሊግ> “ኮከብ ግብ አግቢ” <ብሄራዊ ቡድን> + <ተጫዋች> + <ነጥብ>
		<ዘመን> + <የሊግ ዙር> + <ሀገር> + <ዋንጫ> “ኮከብ ግብ አግቢ” <ቡድን> + <ተጫዋች> + <ነጥብ>
		<ዘመን> + <የሊግ ዙር> + <ሀገር> + <ዋንጫ> “ኮከብ ግብ አግቢ” < ቡድን> + <ተጫዋች> + <ነጥብ>

Annex F: A questionnaire used to collect real football concepts instances and their properties to populate the ontology

1. የ ሰታድየም ስም ዝርዝር
2. የ ኮንፌደሬሽኖች ስም ዝርዝር
3. የፌደሬሽኖች ስም ዝርዝር
4. ለክለቦች ለተጫዋቾች ለአሰልጣኞች ለዳኞች እና ለሌሎችም የሚሰጡ የሽልማት አይነቶች
5. በጨዋታ ጊዜ የዳኞች ሃላፊነት ለምሳሌ ማስጠንቀቂያ መስጠት
6. ኢትዮጵያ ዉስጥ የሚገኙ ዳኞች መረጃ

ተራ ቁጥር	የዳኛዉ ስም	የትውልድ ቀን	የትውልድ ቦታ	ዜግነት	ዖታ	ደሞዝ			
1									

7. ኢትዮጵያ ዉስጥ በ ክለብ ደረጃ የሚካሄዱ የ እግር ኳስ ውድድሮች መረጃ

ተራ ቁጥር	የዉድድር አይነት	በየስንት አመቱ	የትኞቹ ክለቦች	ስንት ዙር አለው					

		ይካሄዳል	ይሳተፋሉ						
1									

8. በጨዋታ ጊዜ በተጫዋቾች የሚከሰቱ ክስተቶች ለምሳሌ ኳስ ማሳለፍ ካስ መሊጋት ኳስ ማንጠር ኳስ ወደ ውጪ ማዉጣት

9. ኢትዮጵያ ዉስጥ የሚግኙ ክለቦች መረጃ

ተራ ቁጥር	ክለብ	ዋና አሰልጣኝ	ረዳት አሰልጣኝ	የክለብ ባለቤት	ስራ አስኪያጅ	ስለ ክለቡ ደጋፊዎች መረጃ ለምሳሌ ብዛታቸዉ	የክለቡ ስፖንሰር		
1									

10. ኢትዮጵያ ዉስጥ በክለብ ደረጃ የታቀፉ ተጫዋቾች መረጃ

ተራ ቁጥር	የ ተጫዋች ስም	የትውልድ ቀን	የትውልድ ቦታ	ዜግነት	ዖታ	የሚሰለፍበት ቦታ	የሚጫችበት ቡድን	ደሞዝ			
1											

[በእያንዳንዱ ጥያቄ ስር ያልተጠቀሰ ነገር ካለም ይፃፉ]

Annex G: The queries and their corresponding relevant document numbers

Query Number	Query	Relevant Documents
Query 1	የኢትዮጵያ ብሄራዊ ቡድን በናሚቢያ 1ለ0 ሲሸነፍ ግቡን ያስቆጠረው ተጫዋች	sportNews59, sportNews61 sportNews18
Query 2	የ1999ዓ.ም የፊፋ አለም አቀፍ ስፖርታዊ ጨዋነት ቀን	sportNews51, sportNews55
Query 3	በ17ኛው ሳምንት የኢትዮጵያ ፕሪምየር ሊግ የደረጃ ሰንጠረዡን ማን ይመራል	sportNews122
Query 4	የኢትዮጵያ ታዳጊ ብሄራዊ ቡድን ከታንዛንያ ጋር ያደረገው የመልስ ጨዋታ ውጤት	sportNews99
Query 5	በ30ኛው አላሙዲን ሲንየር ቻሌንጅ ካፕ ኢትዮጵያ ና ጅቡቲ ብሄራዊ ቡድን ስንት ለ ስንት ተለያዩ	sportNews106, sportNews85, sportNews86
Query 6	በ 1999ዓ/ም የኢትዮጵያ ብሄራዊ ቡድን አሰልጣኝ ዲያን ጋርዚያቶ ረዳት አሰልጣኝ ስም	sportNews32, sportNews86, sportNews101
Query 7	የኢትዮጵያ ብሄራዊ ቡድን እና የቱኒዚያ ብሄራዊ ቡድን የወዳጅነት ጨዋታ ዉጤት	sportNews123, sportNews133,
Query 8	አቶ ሰውነት ቢሻው ለ ዘንድሮው የአፍሪካ ዋንጫ በአጥቂ ክፍል እነማንን አሰለፉ	sportNews124, sportNews133

Query 9	አዲስ አበባ ላይ በተደረገው የወዳጅነት ግጥሚያ የኢትዮጵያ ብሄራዊ ቡድን እና የኒጀር ብሄራዊ ቡድን ስንት ለ ስንት ተለያዩ	sportNews126, sportNews125, sportNews133
Query 10	የኢትዮጵያ ብሄራዊ ቡድን እና የቡርኪና ፋሶ ግጥሚያ	sportNews134, sportNews135, sportNews133
Query 11	በጆሀንስበርግ ብሔራዊ ስታዲየም የተደረገዉ የ29ኛው የአፍረካ ዋንጫ የመክፈቻ ስነስረዓት	sportNews130
Query 12	በ29ኛው የአፍሪካ ዋንጫ የደቡብ አፍሪካ እና የኬቨርድ ብሄራዊ ቡድን ጨዋታ	sportNews131
Query 13	በዘንድሮው የአፍሪካ ዋንጫ የሞሮኮ እና የአንጎላ ብሄራዊ ቡድን የ መጀመርያ ጨዋታ	sportNews131, sportNews137
Query 14	አፍሪካ ዋንጫ ምድብ ሁለት ግጥሚያ	sportNews136, sportNews132 sportNews131
Query 15	የጋና እና ዲሞክራቲክ ኮንጎ ብሄራዊ ቡድን ግጥሚያ ዉጤት	sportNews132, sportNews131
Query 16	ማሊ ብሄራዊ ቡድን ኒጀርን እንድትረታ ያደረጋትን ግብ ያስቆጠረው ማነዉ	sportNews132, sportNews125, sportNews126
Query 17	በ ዋሊያዎች እና በ ዛምቢያ ጨዋታ ላይ የአቶ ሰውነት ቢሻው እና የ ሃርቬ ሬይናርድ መግለጫ	sportNews133, sportNews134
Query 18	በ አፍሪካ ዋንጫ ኢትዮጵያ እና የቺፖሎፖሎዎች የመጀመርያ ግጥሚያ	sportNews134, sportNews135, sportNews133, sportNews128
Query 19	ሰንደይ ኦሊሴ ስለ ዋሊያዎች የሰጠዉ ግላዊ መግለጫ	sportNews135
Query 20	የ ኤለፓንቶች የግጥሚያ	sportNews136
Query 21	በዘንድሮው የአፍሪካ ዋንጫ ያልተሳተፉ ሀገሮች	No answer
Query 22	የ28ኛው የአፍሪካ ዋንጫ ሻምፒዮን	sportNews133
Query 23	የባፋና ባፋና አሰልጣኝ	sportNews131, sportNews134
Query 24	ዋሊያዎች ለሚቀጥለዉ አመት የአለም ዋንጫ የሚያደርጉት ማጣርያ ግጥሚያዎች	sportNews119
Query 25	የሉሲ ወቅታዊ አቋም	No answer

Annex H: The results returned by both classical IR and the proposed system for all the queries

Query	Retrieved Docs (Top 5)	
	The answers returned by the proposed system for each of the queries	**The answers returned by the traditional system for each of the queries**
Query 1	sportNews59.txt sportNews18.txt sportNews61.txt	sportNews61.txt sportNews125.txt sportNews121.txt

	sportNews134.txt sportNews133.txt	sportNews58.txt sportNews59.txt
Query 2	sportNews21.txt sportNews55.txt sportNews51.txt sportNews42.txt sportNews52.txt	sportNews55.txt sportNews51.txt sportNews73.txt sportNews87.txt sportNews42.txt
Query 3	sportNews122.txt sportNews0.txt sportNews52.txt sportNews116.txt sportNews20.txt	sportNews0.txt sportNews122.txt sportNews116.txt sportNews98.txt sportNews110.txt
Query 4	sportNews99.txt sportNews9.txt sportNews48.txt sportNews86.txt sportNews106.txt	sportNews99.txt sportNews98.txt sportNews86.txt sportNews106.txt sportNews126.txt
Query 5	sportNews86.txt sportNews106.txt sportNews85.txt sportNews107.txt sportNews101.txt	sportNews106.txt sportNews85.txt sportNews32.txt sportNews21.txt sportNews133.txt
Query 6	sportNews101.txt sportNews86.txt sportNews32.txt sportNews9.txt sportNews135.txt	sportNews86.txt sportNews32.txt sportNews101.txt sportNews133.txt sportNews99.txt
Query 7	sportNews127.txt sportNews123.txt sportNews133.txt sportNews125.txt	sportNews123.txt sportNews125.txt sportNews126.txt sportNews127.txt

	sportNews126.txt	sportNews133.txt
Query 8	sportNews124.txt sportNews133.txt sportNews134.txt sportNews50.txt sportNews13.txt	sportNews133.txt sportNews124.txt sportNews125.txt sportNews13.txt sportNews50.txt
Query 9	sportNews126.txt sportNews127.txt sportNews125.txt sportNews133.txt sportNews124.txt	sportNews126.txt sportNews125.txt sportNews127.txt sportNews133.txt sportNews119.txt
Query 10	sportNews134.txt sportNews135.txt sportNews133.txt sportNews126.txt sportNews120.txt	sportNews133.txt sportNews126.txt sportNews55.txt sportNews125.txt sportNews134.txt
Query 11	sportNews130.txt sportNews63.txt sportNews133.txt sportNews118.txt sportNews138.txt	sportNews130.txt sportNews129.txt sportNews131.txt sportNews48.txt sportNews128.txt
Query 12	sportNews131.txt sportNews137.txt sportNews134.txt sportNews133.txt sportNews118.txt	sportNews131.txt sportNews129.txt sportNews128.txt sportNews137.txt sportNews130.txt
Query 13	sportNews131.txt sportNews137.txt sportNews133.txt sportNews118.txt sportNews117.txt	sportNews131.txt sportNews137.txt sportNews129.txt sportNews133.txt sportNews128.txt

Query 14	sportNews136.txt sportNews132.txt sportNews9.txt sportNews131.txt sportNews133.txt	sportNews119.txt sportNews131.txt sportNews137.txt sportNews133.txt sportNews128.txt
Query 15	sportNews132.txt sportNews131.txt sportNews34.txt sportNews86.txt sportNews66.txt	sportNews132.txt sportNews131.txt sportNews101.txt sportNews83.txt sportNews58.txt
Query 16	sportNews132.txt sportNews125.txt sportNews126.txt sportNews133.txt sportNews127.txt	sportNews132.txt sportNews131.txt sportNews125.txt sportNews126.txt sportNews61.txt
Query 17	sportNews133.txt sportNews134.txt sportNews128.txt sportNews126.txt sportNews86.txt	sportNews133.txt sportNews88.txt sportNews125.txt sportNews127.txt sportNews78.txt
Query 18	sportNews133.txt sportNews128.txt sportNews10.txt sportNews134.txt sportNews135.txt	sportNews125.txt sportNews128.txt sportNews133.txt sportNews119.txt sportNews126.txt
Query 19	sportNews135.txt sportNews134.txt sportNews133.txt sportNews126.txt sportNews120.txt	sportNews135.txt sportNews139.txt sportNews101.txt sportNews97.txt sportNews127.txt
Query 20	sportNews136.txt	sportNews129.txt

	sportNews34.txt sportNews86.txt sportNews66.txt sportNews88.txt	sportNews4.txt sportNews75.txt sportNews131.txt sportNews28.txt
Query 21	sportNews117.txt sportNews134.txt sportNews131.txt sportNews135.txt sportNews118.txt	sportNews42.txt sportNews117.txt sportNews138.txt sportNews118.txt sportNews131.txt
Query 22	sportNews133.txt sportNews118.txt sportNews117.txt sportNews134.txt sportNews10.txt	sportNews120.txt sportNews135.txt sportNews108.txt sportNews42.txt sportNews7.txt
Query 23	sportNews131.txt sportNews120.txt sportNews134.txt sportNews101.txt sportNews19.txt	sportNews13.txt sportNews50.txt sportNews32.txt sportNews19.txt sportNews101.txt
Query 24	sportNews119.txt sportNews120.txt sportNews134.txt sportNews63.txt sportNews133.txt	sportNews19.txt sportNews131.txt sportNews15.txt sportNews43.txt sportNews74.txt
Query 25	sportNews118.txt sportNews133.txt sportNews134.txt sportNews136.txt sportNews135.txt	sportNews133.txt sportNews54.txt sportNews127.txt sportNews125.txt sportNews128.txt

Annex I: The precision, recall, and F-measure values for each of the queries for both classical IR and the proposed system based on the initial expert judgment

Query number	The proposed system			classical IR		
	Precision	Recall	F-measure	Percision	Recall	F-measure
Query 1	0.6	1	0.75	0.4	0.66	0.5
Query 2	0.4	1	0.57	0.4	1	0.57
Query 3	0.2	1	0.33	0.2	1	0.33
Query 4	0.2	1	0.33	0.2	1	0.33
Query 5	0.6	1	0.75	0.4	0.66	0.5
Query 6	0.6	1	0.75	0.6	1	0.75
Query 7	0.4	1	0.57	0.4	1	0.57
Query 8	0.2	1	0.33	0.2	1	0.33
Query 9	0.6	1	0.75	0.4	1	0.57
Query 10	0.6	1	0.75	0.4	0.66	0.5
Query 11	0.2	1	0.33	0.2	1	0.33
Query 12	0.2	1	0.33	0.2	1	0.33
Query 13	0.4	1	0.57	0.4	1	0.57
Query 14	0.6	1	0.75	0.2	0.33	0.25
Query 15	0.4	1	0.57	0.4	1	0.57
Query 16	0.6	1	0.75	0.6	1	0.75
Query 17	0.4	1	0.57	0.2	0.33	0.25
Query 18	0.8	1	0.88	0.4	0.5	0.44
Query 19	0.2	1	0.33	0.2	1	0.33
Query 20	0.2	1	0.33	0	0	0
Query 21	0	0	0.33	0	0	0
Query 22	0.2	1	0.33	0	0	0
Query 23	0.4	1	0.57	0	0	0
Query 24	0.2	1	0.33	0	0	0
Query 25	0	0	0.33	0	0	0

Annex J: The precision, recall, and F-measure values for each of the queries for both classical IR and the proposed system based on the refined expert judgment

Query number	The proposed system			classical IR		
	Precision	Recall	F-measure	Percision	Recall	F-measure
Query 1	0.6	1	0.75	0.4	0.66	0.5
Query 2	0.4	1	0.57	0.4	1	0.57
Query 3	0.4	1	0.57	0.4	1	0.57
Query 4	0.2	1	0.33	0.2	1	0.33
Query 5	0.6	1	0.75	0.4	0.66	0.5
Query 6	0.6	1	0.75	0.6	1	0.75
Query 7	1	1	1	1	1	1
Query 8	0.4	1	0.57	0.2	1	0.33
Query 9	0.8	1	0.88	0.6	1	0.75
Query 10	0.6	1	0.75	0.4	0.66	0.5
Query 11	0.2	1	0.33	0.2	1	0.33
Query 12	0.2	1	0.33	0.2	1	0.33
Query 13	0.4	1	0.57	0.4	1	0.57
Query 14	0.6	1	0.75	0.2	0.33	0.25
Query 15	0.4	1	0.57	0.4	1	0.57
Query 16	0.6	1	0.75	0.6	1	0.75
Query 17	0.4	1	0.57	0.2	0.33	0.25
Query 18	0.8	1	0.88	0.4	0.5	0.44
Query 19	0.2	1	0.33	0.2	1	0.33
Query 20	0.2	1	0.33	0	0	0
Query 21	0.2	1	0.33	0	0	0
Query 22	0.2	1	0.33	0	0	0
Query 23	0.4	1	0.57	0	0	0
Query 24	0.2	1	0.33	0	0	0
Query 25	0.2	1	0.33	0	0	0

Declaration

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

Declared by:

Name: **Genet Asefa Gesese**

Signature: _____

Date: _____

Confirmed by advisor:

Name: **Dr Fekade Getahun**

Signature: _____

Date: _____

Place and date of submission: Addis Ababa University, March, 2013.