ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
COLLEGE OF NATURAL SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

Automatic Ontology Learning From Unstructured Amharic Text

BY

BERHANU MENGISTE

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 SCIENCE IN COMPUTER SCIENCE

March, 2013

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
COLLEGE OF NATURAL SCIENCES
DEPARTMENT OF COMPUTER SCIENCE

Automatic Ontology Learning From Unstructured Amharic Text

BY

BERHANU MENGISTE

ADVISOR: FEKADE GETAHUN (PhD)

Signature of the board of examiners for Approval

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

DEDICATED TO:

My mother ***Aregash Legesse(Abaye)***

My father ***Mengiste Ketema (Gashe)***

My brothers and sisters

Acknowledgements

It is with sincere gratitude that I thank my advisor, Dr Fekade Getahun, for his critical comments, persistent encouragement and advice on every of the steps during this research.

It is also with unadulterated love and respect that I thank all my family for being on my side through all this time.

Mr. Solomon, a research student at Addis Ababa University. Your comments were important in the linguistic Sections of my research. And I thank you and your friends for their cooperation to evaluate my tools.

Mr. Tessema Mindaye thank you for providing me the stop words list, the stop word remover and the lemmatizer, which were very helpful. Thank you very much.

Last but not least I would like to thank all my friends, Bese, Frech, Bini, Tare, Addise, Neway, Anbes, Mule, Des, Addioch, Geni, Abby, Hone for your unconstrained support and true friendship. All friends, including those not mentioned, I will be missing you.

Table of contents

LISTOFTABLES

LIST OF FIGURES

ABBREVIATIONS

AOL: Amharic Ontology Learner

TF-IDF: Term Frequency – Inverted Document Frequency

ACE: Automatic Concept Extractor

POS: Parts Of Speech

NLP: Natural Language Processing

POM: Probabilistic Ontology Modeling

AI: Artificial Intelligence

ABSTRACT

This research proposes a method, Amharic ontology learner, which helps to automatically learn or extract ontology from an unstructured Amharic text. Amharic ontology learner handles the ontology learning process through distinct process layers, concept extraction, taxonomy building, and non-taxonomic relations mining.

Once all potential concepts are extracted a concept hierarchy (taxonomy) is formed, which is then supplemented by non-taxonomic relations to evolve the taxonomy into a full ontology. Different methods have been used to implement each layer.

Amharic ontology learner is based on both single-word and multi-word concepts, as these make the ontology to be represented by a more solid and distinctive concepts. A hierarchical agglomerative clustering method is used for building the domain taxonomy. To identify the non-taxonomic relations a linguistic method, verbal expressions as a relation indicator, is used and a method which tries to find out the most appropriate level of generalization for the relation is also implemented at the top of the non-taxonomic relation mining module.

To practically test the performance of the methods, modules in Amharic ontology learner are implemented. Our method can also represent the extracted ontology in OWL using Jena Semantic Web Framework. Amharic ontology learner is applied to an already tagged news corpus from WALTA News Agency. The result shows that Amharic ontology learner can be used as a starting point for future researches related to Ontologies and Ontology learning from Amharic text.

Keywords: Ontology, Ontology learning, Concept, taxonomy, Concept relationship.

CHAPTER ONE: INTRODUCTION

1.1. BACKGROUND

Ontologies constitute a formal conceptualization of a particular domain of interest that is shared by a group of people. Following T. Berners-Lee's [46] vision of the Semantic Web at the beginning of the twentieth century, ontology became a core solution to many problems concerning the fact that computers do not understand human language. If there were ontology and every document were marked up with it and we had agents/computers that would understand the ontology, then computers would finally be able to process our requests in a really fashionable way.

Ontologies can facilitate text understanding and automatic processing of textual resources. Moving from words to concepts not only eases data sparseness issues, but also gives interesting solutions to polysemy and homonymy by filtering out non-ambiguous concepts that may map to various realizations in possibly ambiguous words [2].

Ontologies have been applied to a number of different domains, including biomedicine [6], finance, education [8] and software engineering [7]. Due to these and other sound applications different methodologies are proposed by different researchers for the design and building of ontologies. Manual ontology construction process is time consuming and cumbersome. Hence, the design and development of methods and software tools to support automatic ontology construction is an important research topic, which is known as ontology learning [3].

Ontologies formalize the intentional aspects of a domain, whereas the extensional part is provided by a knowledgebase that contains assertions about instances of concepts and relations as defined by the ontology. The process of defining and instantiating a knowledgebase is referred to as knowledge markup

or ontology population, whereas (semi-)automatic support in ontology development is usually referred to as ontology learning.

Ontology learning is concerned with knowledge acquisition and in the context of this research more specifically with knowledge acquisition from text. Obviously, much of the work in this area builds on the large body of work in this direction within NLP, AI, and machine learning. As such, the legitimate question arises if the wheel is not being reinvented. Is ontology learning simply a revision of existing ideas and techniques under a new name? The answer to this should be: no. Although the aims of knowledge acquisition and ontology learning (from text) are certainly overlapping, in essence the acquisition of explicit knowledge implicitly contained in (textual) data, there are however, a number of novel and innovative aspects to ontology learning that sets it apart from much of the previous work in knowledge acquisition:

- Ontology learning is inherently multidisciplinary due to its strong connection with the Semantic Web, which has attracted researchers from a very broad variety of disciplines: knowledge representation, logic, philosophy, databases, machine learning, natural language processing, etc. In consequence, ontology learning has profited from a massive exchange of ideas and techniques that shaped a somewhat different vision of the knowledge acquisition problem.
- Ontology learning, in the Semantic Web context, is primarily concerned with knowledge acquisition from and for Web content and is thus moving away from small and homogeneous data collections to tackle the massive data heterogeneity of the World Wide Web instead.
- Given the machine learning origins of much of the work in ontology learning, the field is rapidly adapting the rigorous evaluation methods that are central to most machine learning work. Therefore, ontology learning will be impacted by efforts to systematically evaluate and compare approaches on well-defined tasks and with well-defined

evaluation measures, thus making it a highly challenging field in which only competitive and demonstrable approaches will survive.

In summary, these aspects indeed establish ontology learning as a new and challenging area in its own right, with a lot of innovating research to which also this thesis hopes to contribute.

As it could be expected the majority of ontology learning methodologies and tools are designed for English, some have tried to have it in Spanish [1] but in recent days the number of Amharic documents on the web is increasing. In addition, adopting ontology learning methods and tools as they are to a local context is hardly possible because Amharic texts have their own peculiar syntax and grammar[20]. Hence, designing methods and tools that can work with Amharic text is a must, which is the core of this research.

1.2. STATEMENT OF THE PROBLEM

Ontologies serve as a means for knowledge representation and are capable of expressing a set of entities, their relationships, constraints, axioms and the vocabulary of a given domain. However, the manual construction of ontologies is an expensive and time consuming task because the professionals required for this task (i.e. a domain specialist and a knowledge engineer) usually are highly specialized. This difficulty in capturing the knowledge required by knowledge based systems is becoming very common and is labeled as "knowledge acquisition bottleneck".

Hence, fast and cheap ontology development is crucial for the success of knowledge based applications and the Semantic Web. This has been designed and implemented for English [4] and Spanish [1].

Amharic is written with a version of the Ge'ez script known as ፊደል(Fidel) and has its own unique grammar, syntax, character (fidel) representation and statement formation [10].

These reasons make it clear that ontology learning from Amharic text involves reasonably different methods and tools compared to English or Spanish text. To the best of the researcher's knowledge no such tools exist for Amharic text.
Hence, in this research the problems to be addressed are:
What are the approaches, methods and tools that facilitate the automatic learning of ontology from Amharic text? And how these methods and tools can be realized?

1.3. OBJECTIVES

1.3.1. GENERAL OBJECTIVE

The general objective of this research work is to propose methods for automatic ontology learning from Amharic text and design tools that can perform the learning process.

1.3.2. SPECIFIC OBJECTIVES

The specific objectives of this research work are:

1.1. To propose approach/algorithm that extract concepts from Amharic document.
1.2. To mine relationships existing between extracted concepts.
1.3. To construct the resulting Ontology.
1.4. To evaluate the performance and relevance of the proposed algorithm.

1.4. SCOPE AND LIMITATIONS OF THE STUDY

The first step in ontology learning is concept extraction, which requires basic NLP operations including sentence splitting, tokenising, parts of speech tagging, lemmatizing and named entity recognising. However, most of these tools are unavailable except the Lemmatizer. This lack of required tools obliged us to design most of the tools by our own such as, sentence splitting and POS recognizer from tagged corpus. Our inability to find Amharic parts of speech tagger has also tightened our option to choose comfortable corpus and we are

forced to use an already tagged news corpus. This obviously took away our advantage to test our methods on different domain specific corpus.

The scope of this research is limited to constructing ontology from the News data collected from WALTA. The Amharic texts corpus, the input, will be from unstructured file (documents), not from databases or semi structured files.

1.5. METHODOLOGY

This research is conducted by first reviewing a number of related literatures. Python 3.1/2.7 and java programming languages are used to develop the prototype tools for different tasks.

Combinations of statistical and linguistic approaches are employed during this research, which help us get all the benefits from both the approaches. C-value [21] and TF-IDF [25] methods which are used during concept extraction phase are statistical whereas linguistic filtering and non-taxonomic relations mining techniques used are more of linguistic in genre. Hierarchical agglomerative clustering technique is employed for taxonomic relationship mining [47]. Verbal expressions in a sentence are considered as source of information to find non-taxonomic relations among concepts in that specific sentence. Jena semantic web framework 2.4[1]is used to generate the ontology in OWL[2] and it can be viewed using protégé 4.2[3].

The performance of the methods and the tools is measured using one of the main scoring approaches: precision [48].

Recall is not used as a measure because it is not feasible to apply it in this research. Because recall requires to manually acquire all true results or needs a reference system that its results are thought to be ideally correct. However, both are not available in our case, because there is no Amharic ontology

[1] *Jena*™ is a Java *framework* for building *Semantic Web* applications: http://jena.apache.org/
[2] OWL is a W3C standard for ontology representation: http://www.w3.org/standards/techs/owl#w3c_all
[3] Protege is a Java tool providing an extensible architecture for the creation of knowledge-based applications: ***http://protege.stanford.edu/***

learner and doing the extraction manually is yet impossible. Precision and recall are most commonly used for the assessment of information retrieval systems and trace their origins back to the information retrieval discipline.

The evaluation is made for the three major tasks of this research; that are *concept extraction*, *taxonomy building*, *non-taxonomic relations mining*.

1.6. SIGNIFICANCE OF THE RESEARCH

As it has been stated in the scope this research is limited to constructing ontology from unstructured text. It is known that manually creating ontology is not a welcomed approach these days due to the fact that it is computationally expensive and the increase in size of the web content from which an ontology is to be learned. This research will surely simplify this hard job and the resulting ontology can be an input for different disciplines as:

- Knowledge representation/sharing [15].
- Semantic Digital Libraries [13].
- Software engineering [7].
- Multi-agent systems [14].
- Ontology based reasoning [16].

1.7. ORGANIZATION OF THE THESIS

This document contains six chapters including this chapter. Chapter two and chapter three presents review of related literatures and related works respectively. In chapter four design and implementation of methods for Amharic ontology learner are presented. Chapter five has the evaluation and results and in the final chapter conclusion and recommendations are given.

CHAPTER TWO: LITERATURE REVIEW

In this chapter we review existing techniques and methods for automatic learning of ontology from text. The review will begin by discussing issues related to ontologies, ontology languages and components. Then we will look into most commonly used methods and tools for handling sub-tasks in ontology learning (term extraction, taxonomy building and relationship mining).

2.1. ONTOLOGIES

Ontology has its origin in philosophy and is defined as the branch of philosophy which deals with the nature and the organization of reality [2]. In computer science and information science, Ontology formally represents knowledge as a set of concepts within a domain, and the relationship between those concepts. It is also introduced by T.Berners-Lee [49] as a means to represent data in semantic web in his book, weaving the web as follows:

"*The vision of the Semantic Web is to extend principles of the Web from documents to data; data should be related to one another just as documents (or portions of documents) are already. This also means creation of a common framework that allows data to be shared and reused across application, enterprise, and community boundaries, to be processed automatically by tools as well as manually, including revealing possible new relationships among pieces of data*"

2.1.1. ONTOLOGY LANGUAGES

Ontology languages are formal languages used to construct ontologies. They allow the encoding of knowledge about specific domains and often include reasoning rules that support the processing of that knowledge. Usually Ontology languages are declarative in genre.

A considerable effort is exerted in building ontology languages, which facilitates the creation of a common framework that allows data to be shared more

efficiently between applications. That means, a specific domain could now be described in terms of structured data, with the use of an ontology language. Examples of these languages are RDF - RDF(S), OIL, DAML+OIL, the most recent and common one is OWL. OWL is the web ontology language, developed by the W3CWeb Ontology (WebOnt) Working Group. OWL is mainly based on OIL and DAML+OIL, and therefore the main features of OWL are very similar to those of OIL.OWL was proposed as a W3C recommendation in February 2004 [26].

OWL will be our preferred language for representing ontology because it is assumed to be better than the others for the following reasons.

OWL incorporates expressions for versioning. OWL has a rich expressive power and possesses a layered architecture for scalability. OWL supports different natural languages compared to the other considered languages. These days OWL is very well positioned in the community mobilizing lots of efforts to make it become the Semantic Web language of the future.

OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary. OWL can be used to represent the meaning of terms and the relationships between those terms. It is more expressive than XML, RDF and RDF-S, making it easier to represent machine interpretable content on the web. OWL has three species:

- **OWL Lite**: OWL Lite was originally intended to support those users primarily needing a classification hierarchy and simple constraints. OWL Lite excludes enumerated classes, disjointness statements and arbitrary cardinality. The advantage of this species is that, it is both easier to grasp (for users) and easier to implement (for tool builders). The disadvantage is of course a restricted expressivity [26].

- **OWL DL**: OWL DL was designed to provide the maximum expressiveness possible while retaining computational. OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, number restrictions may not be placed upon properties which are declared to be transitive). OWL DL is so named due to its correspondence with description logic, a field of research that has studied the logics that form the formal foundation of OWL [26].
- **OWL Full**: The entire language is called OWL Full, and uses all the OWL language primitives. It also allows combining these primitives in arbitrary ways with RDF and RDF Schema.OWL Full is based on a different semantics from OWL Lite or OWL DL, and was designed to preserve some compatibility with RDF Schema. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right; this is not permitted in OWL DL. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary [26].

2.1.2. COMPONENTS OF ONTOLOGY

Ontologies are known to have different components. Though there are different doctrines for ontology implementation, they usually have the following components in common.

Classes: Are concepts within a domain. They are also known to be the backbones of the ontology. Concepts that are extracted from a corpus are usually considered as classes, which appear to be a node in a taxonomic tree.

Relations: Tell us how concepts (usually two) are related or associated with each other. Relations are usually considered as a directed arrow in a graph where the source of arrow (the first argument in a relation) is said to be the domain and the destination of the arrow (the second argument in a relation) is referred to as a range. For example, the binary relation subclass-of is used for

building the class taxonomy: primary school (የመጀመሪያ ደረጃ ትምህርት ቤት) is a subclass of school (ትምህርት ቤት). In the relation Subclass-Of (የመጀመሪያ ደረጃ ትምህርት ቤት, ትምህርት ቤት), the domain is primary school (የመጀመሪያ ደረጃ ትምህርት ቤት) whereas school (ትምህርት ቤት) is the range of the relation. Besides taxonomic relations, there are also non-taxonomic relations which are crucial for fully representing the domain in question. Learning of a relation includes identifying the relation label (text describing a relation) and the appropriate domain and range of a relation.

In ontology, there exist more complex relation types between concepts or attributes that are called axioms, which may describe the properties of a relation, such as transitivity or symmetry and disjointness or equivalence properties for concepts.

2.2. THE ONTOLOGY DEVELOPMENT PROCESS

In this sub-section we will discuss the process, sequence of tasks that constitute the overall ontology development process. As any other complex process it is an agreed and buoyant idea to divide the whole task into manageable and organized units, so that the overall process is clear and maintainable.

Buitelaar and Cimiano [17] suggested that phases in ontology learning can be organized into a layer stack, Figure 2.1, and the ontology learning process always precedes bottom up according the layer stack.

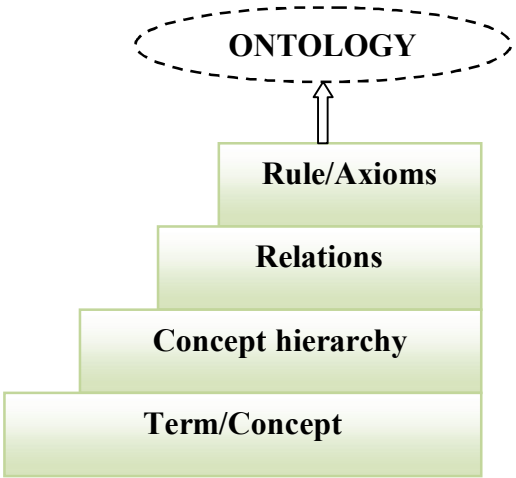

Figure 2.1: Ontology learning layer cake

The higher layers rely on the output of the lower layers. For example, we cannot discover relations between concepts before potential concepts are available from the extraction phase. According to this layer structure, term extraction is the first step during ontology construction, without potential terms being recognized one can't move a step ahead.

The concept hierarchy layer is the second and the most important one in ontology learning process. It happens to be a collection of extracted concepts organized into a hierarchical structure. If we can assume the structure to be a tree; the concepts can be considered as a node that can be characterized by attributes as well as by their relationship with other concepts.

Non- taxonomic relations are extracted at the step next to concept hierarchy generation, these relations are mostly domain specific and difficult to find but also the most important ones to make the resulting ontology complete.

In the last layer, axioms or rules are defined. Formal axioms serve to model sentences that are always true. They are normally used to represent knowledge that cannot be formally defined by the other components. In addition, formal axioms are used to verify the consistency of the ontology itself.

2.3. NATURAL LANGUAGE PROCESSING

Ontology is to be learnt from unstructured text which is not collected to be used for ontology learning, so a pre-processing on a corpus is a must. In order to make the corpus suited for processing, the text collection must pass through several steps. A corpus preprocessing procedure may consist of the following sub tasks:

- **Sentence Splitting**: This is the step where sentences are separately recognized from a text.
- **Tokenization**: The process by which a text is broken into its constituent tokens. Tokenization can occur at different levels: a text could be broken up into paragraphs, sentences, words.

- **Part-of-Speech (POS) tagging**: It is the task of assigning to each token its corresponding syntactic word category (Part-of-speech, i.e. noun, verb, adjective etc).
- **Lemmatization / Morphological Analysis**: This is a step where morphological variants of words are reduced into their corresponding base form/lema/root. For example the word "የዋልት" becomes "ዋልት" or the word "ህዝቦች" becomes "ህዝብ".

2.4. CONCEPT EXTRACTION

In this sub-section, we discuss methods used in extracting concepts. Terms (linguistic representation for concepts) can be either a single word or multi words.

2.4.1. APPROACHES TO AUTHOMATIC CONCEPT EXTRACTION (ACE)

Different approaches to ACE have been suggested by different researchers with the aid of available language resources and NLP technologies. Methodologically, they can be classified into three categories: linguistic, statistical and hybrid approaches.

2.4.1.1. LINGUISTIC APPROACH

The linguistic approach basically tries to identify terms considering their syntactic properties, mostly the Parts Of Speech (word class) of words and sequence of words. In fact, it has been proved that terms usually have characteristic syntactic structures, called synaptic compositions, which tell us about the term's properties. For example, Sphia Ananiadou[23]in his work on automatic term recognition studied the morphological aspects of term formation and showed by evidence that linguistic knowledge is important and even makes the term recognition process effective.

There was also a research by Bourigault [24], surface grammatical analysis for term recognition, which tries to extract terminological noun phrases through a

deep study on the grammatical structure of the input text. These researchers have also tried to identify term candidates in a part-of-speech (POS) tagged corpus by a predefined POS pattern, e.g., Noun + Noun, (Adj| Noun) + Noun or((A|N) | ((A|N) (NP)) (A|N)) N. Whenever term recognition through linguistic approach is thought POS tagging is the common and a pre-requisite for any further syntactic processing or analysis. And then, depending on the language's grammatical rules syntactic patterns that are assumed to represent a term (e.g. Noun – Noun phrases) are applied and those that match none of the patterns will be filtered out.

This approach was reported to achieve good results, in particular, on small scale corpora [23, 24]. However, it sometimes suffer from inherent disadvantages due to the drawback of the pre-defined syntactic patterns, these include(1) not including all possible patterns in which a term may occur, (2) Unreliable applicability to other domains or languages, and (3) incapability of telling apart the true terms and non-terms of the same structure.

Note that not all candidates consistent with predefined structural patterns are true terms, while not all others are necessarily non-terms. It can be easily thought that not all noun-noun compounds are true terms. Thus, a basic limitation with this approach is that why a pre-defined structural pattern is given all the trust to tell whether a term candidate truly represents a domain-specific concept or not.

2.4.1.2. STATISTICAL APPROACH

This approach attempts to use statistical information to recognize true terms from a document. The most commonly used statistical information is frequency, and the simplest way to find this information from a corpus is through counting. However, frequency itself alone is not sufficient criterion for deciding that a term is a true term or not. The un-convincing downside with a statistical approach solely relying on frequency information is that, on the one hand, most of the frequent or common words in a language are function

words or stop words instead of concrete true terms and, on the other hand, infrequent words are not necessarily useless, in view of their rare use, to represent a subject field as the true terms do.

Different statistical information has been used to make easy the multi-word ACE; following the assumption that a multi-word term carries a key concept and is thus expected to behave like an atomic text unit, various statistical measures are applied to explore such unity or unithood[4]. Among them the popular ones are *log-likelihood ratio* [28], *C-value* and *NC-value* [21] and *imp* function [29].

Usually, a statistical measure of this kind is used in combination with some syntactic patterns, as it can be demonstrated in most hybrid approaches to ACE. The linguistic filtering through the pre-defined patterns is first applied to filter out term candidates, and then a statistical measure is applied to evaluate the true terms among the candidates. For example, the *imp* function [29] is applied only to noun compounds each consisting of a number of simple nouns. It calculates the termhood[5]of a compound candidate in terms of the termhood of its component nouns.

Wermter and Hahn[40]by their research on finding terms from a very large corpora, they investigated the internal stability of a term candidate and identify biomedical multi-word terms among n-grams of words from a large corpus, measuring their termhood in terms of their paradigmatic modifiability P-Mod. The P-Mod of an n-gram is defined as the product of the modifiability of each possible combination of positions (or slots) within the n-gram. A lower P-Mod score indicates a more stable structure and accordingly a higher termhood.

[4]**Unithood:**refers to a degree of strength or stability of syntagmatic combinations or collocations. [22]
[5]**Termhood**refers to a degree of linguistic unit. It considers a term as a linguistic unit representative for the document content.[22]

The main assumption for this termhood measure is that true terms are linguistically more permanent than non-terms. It is also worth noting that being structurally stable is not a sufficient condition for a true term, otherwise all structurally stable text units, e.g., idioms and fixed expressions would be terms in a subject field.

The most obvious merit of a statistical approach is its independency to a domain and language, although its effectiveness can be challenging while working on small corpora. TF-IDF is one of the statistical techniques available.

2.4.1.2.1. TF-IDF ALGORITHM

TF–IDF [25], Term Frequency–Inverse Document Frequency, is a numerical statistic which reflects how important a word is to a document in a document collection or corpus. TF–IDF is the product of two statistical values, term frequency and inverse document frequency. Various ways for determining the exact values of both statistics exist.

In the case of the term frequency, *tf(t,d)*, the simplest choice is to use the raw frequency of a term in a document, i.e. the number of times a term **t** occurs in document **d**. If we denote the raw frequency of **t** by *f*(t,d), then the simple tf scheme is *tf*(t,d) = *f*(t,d).

The inverse document frequency is a measure of whether the term is common or rare across all documents. It is obtained as a ratio of the total number of documents over the number of documents containing the term, and then taking the logarithm of that quotient.

$$idf(t, D) = \log \frac{|D|}{1 + |\{d \in D : t \in D\}|} \quad \text{.....2.1}$$

Where:

- $|D|$: the total number of documents in the corpus.
- $|\{d \in D : t \in D\}|$: The number of documents where the term t appears.

If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the formula to $1 + |\{d \in D: t \in D\}|$.

A high weight in TF–IDF is achieved by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents.

2.4.1.3. HYBRID APPROACH

This approach integrates both linguistic and statistical methods in a way that advantages are maximized and problems are minimized. The linguistic methods usually run first to filter out possible candidates, which then will be forwarded to the statistical methods for measuring the term hood of a string in number.

Daille[39] presents a hybrid approach to ACE that works in the following manner. First a linguistic filtering is conducted to filter out term candidates from a text input, and then, several statistical scores, including *frequency*, *MI,Φ coefficient* and *Log like coefficient*, are applied to produce a final set of terms as output.

C-value and NC-value method which is proposed by Frantzi and Ananiadou[21]works in a considerably similar way. First, linguistic restrictions (e.g., using syntactic patterns) are applied to generate initial candidates. And then, the C-value, which combines a few statistical features of a nested multi-word candidate, and the NC-value, which integrates context information into the C-value, are applied to carry out a finer extraction of multi-word terms from the candidates.

Maynard and Ananiadou[41]show up with a model called TRUCKS to further improve upon the existing statistical approaches by including different contextual information, e.g. semantic and classification information from an external thesaurus.

A hybrid approach can also be applied by joining a number of independent term recognizers together. Vivaldi [42] showed that a simple integration of different term recognition methods will result in a consistent improvement in performance upon any of the component term recognizers alone. In addition, various NLP technologies available may facilitate ACE, e.g., shallow parsing, unknown word detection, and term variation recognition.

This research is intended to use this approach, hybrid approach, because it is wiser to exploit the advantages of both statistical and linguistic approaches. This research uses TF-IDF for single word and C-value method for multi-word concept extraction.

2.4.1.3.1. C-VALUE METHOD

C-value is a measure a measure proposed by Frantzi et al [21] to determine the degree of a term to represent a concept in a certain domain. The C-value is computed with two different formulas depending on the nature of the term; (1) the term is found nested/included in another longer term and (2) the term is not found as part of another longer candidate term.

If the candidate term was never found nested/included in another longer candidate term the C-value calculation is straight forward as shown in equation 2.3. Whereas if it was found as part of another longer candidate term, the calculation needs two extra parameters:

- The frequency of a term as part of longer candidate terms
- The number of those longer candidate terms that incorporated it.

These two parameters are computed as follows: For every string **a**, that is extracted as a candidate string and for each substring **b** of **a**, we compute triples.

(f(b); t(b); c(b)),

Where

- f(b) is the total frequency of **b** in the corpus
- t(b) is the frequency of term **b** nested in another longer candidate terms
- c(b) is the number of these longer candidate terms that host term **b**.

To calculate these triples, first c(b) is initialized to 1 and t(b) is initialized to the frequency of **a**. Each time **b** is found nested in another candidate term **a** c(b) is incremented by 1 and t(b) is incremented by the frequency of the longer candidate term **a. i.e.** f(**a**). The same thing works if **a** is also included in another longer candidate term. f(b) is a simply a raw count of the term b, it needs no further computation. Once c(b) and t(b) are initialized, the iterative computation on c(b) and t(b) can be summarized using equation 2.2.

$$\forall \boldsymbol{b \epsilon a} = \begin{cases} \boldsymbol{C(b)} = \boldsymbol{C(b)} + \mathbf{1} \\ \boldsymbol{t(b)} = \boldsymbol{t(b)} + \boldsymbol{f(a)} \end{cases} \quad \text{..... 2.2.}$$

C-value computation is begun right after these triples are computed for every term that is found nested in another term. So, for terms that are never found in another longer term, their C-value is calculated using equation2.3. Whereas for terms that have their triples computed C-value is calculated using equation 2.4.

$$\boldsymbol{C - calue(a)} = \mathbf{log_2|a|}.\boldsymbol{f(a)} \text{ ..... 2.3}$$

$$\mathbf{C - calue(a)} = \mathbf{log_2|a|}.\mathbf{f(a)} - \frac{\mathbf{1}}{\mathbf{P(Ta)}}\sum_{\mathbf{a \epsilon Ta}} \mathbf{f(a)} \text{ ..... 2.4}$$

Where:

a: is the candidate string,

f(a):is its frequency of occurrence of **a** in the corpus,

T_a: is the set of extracted candidate terms that contain **a**,

$P(T_a)$: is the number of these candidate terms.

2.5. LEARNING TAXONOMIC RELATIONS

In this section, we discuss works related to the automatic acquisition of taxonomies. The main paradigm for learning taxonomic relations exploited in the literature is clustering approaches. There is no single correct class hierarchy for any given domain. The hierarchy may depend on the reason for which the ontology is designed for and it may also depend on the level of the detail that is necessary for the application, even sometimes depends on the need to make it compatible with other models.

One of the first works on taxonomy building through clustering nouns was the one by Hindle [31], in which nouns are clustered into a class according to their extent to appear in similar verb frames. In particular, he takes into account nouns appearing as subjects and objects of verbs.

The work of Faure and Nedellec [33] is based on the distributional hypothesis and they propose an iterative bottom-up clustering technique for nouns that appear in similar contexts. In each step, they cluster the two most similar nouns based on their context. However, their approach requires manual validation after each clustering step.

Caraballo[34] uses clustering methods to derive an unlabeled hierarchy of nouns by using data about conjunctions of nouns from the Wall Street Journal corpus. Interestingly, Caraballo also labels the abstract concepts of the hierarchy in which the children of the concept in question appear as hyponyms. The most frequent hypernym is then chosen in order to label the concept. The final tree was then evaluated by choosing any clusters at random giving it to human experts for evaluation.

Bisson et al. [37] uses bottom-up clustering and compare different similarity/distance metrics as well as different pruning parameters. Regarding the use of lexico-syntactic patterns denoting a certain semantic relation there are a lot more approaches [35,36,38]. Hearst [38] aims at the acquisition of

hyponym relations from Grolier's American Academic Encyclopedia. In order to identify these relations, Heart employed a lexico-syntactic patterns manually acquired from the corpus. The approaches of Hearst and others are characterized by a (relatively) high precision in the sense that the quality of the learned relations is very high. However, these approaches have relatively low recall.

In this research, a hierarchical agglomerative clustering is applied to learn all possible taxonomies from the text, a rule based approach using certain syntactic patterns is also potentially applicable, but will not be considered in this research because a news corpus does not have patterns than we can apply rules onto.

2.6. LEARNING NON-TAXONOMIC REALTIONS

In this section we review existing techniques for automatically extracting non-taxonomic relations from text, non-taxonomic relations can be any domain specific relations that could exist between concepts. In contrast to taxonomic (is-a) relation, which establish abstraction hierarchies, these are relationships which express that one concept is logically related to another. Extracting non-taxonomic relations are comparatively difficult because such relations can't be suggested using external knowledge base and it is also rare to be found from a simple set of syntactic patterns. This is because such relations are domain dependent and occurs inconsistently. Non taxonomic relations are important in this research because those relationships are the ones that enrich our taxonomy and transform it into a full ontology.

Generalizing textual patterns (both manually and automatically) for the identification of relations has been proposed since the early nineties [38], and it has been applied to extending ontology with hypernymy and holonymy relations. The non-taxonomic relations can be found in different methods, but the following two are relevant to be discussed: using association rules and using linguistic criteria.

2.6.1. USING ASSOCIATION RULES

Association rules are commonly used to discover data, text elements or patterns that co-occur frequently within a dataset. Such patterns can be used to make predictions on data. It was first introduced by Agrawal in [43] as a technique for market basket analysis. The aim there was to find association rules that predict the purchasing behavior of customers.

When bringing it to ontology learning, transactions are defined in terms of words occurring together in certain syntactic dependencies. If the rule X=> Y has been generated and stored, we can conclude that there is a relationship between the concepts in X and the concepts in Y.

2.6.2. LINGUISTIC CRITERIA

In this approach the task of learning relations from a text is based on verbal phrases. The main idea lies on the extraction of verb frames. In verbal frames the verbs indicate the relationship between concepts in that clause or statement, as suggested by knowledge engineering researchers [18] and also by Amharic linguistic researchers [19, 20].

Once the relation is found there is also a need to find the proper generalization for that relation. Cimiano[30] works on this and tries to find the proper generalization from the taxonomy. E.g. for the relation suffer_from, the relations suffer_from (older man, head ache), suffer_from (ill person, head ache), suffer_from (woman, stomach ache) are certainly valid. However, from an ontology point of view we are interested in finding the relation at the appropriate level of generalization that can describe the domain in question. So this relation shall be represented at the level appropriate suffer_from (patient, ache). The way he was able to find this generalization level is using a conditional probability technique, to be discussed in Section 4.6.1.

CHAPTER THREE: RELATED WORK

3.1. INTRODUCTION

In the field of ontology learning there are tools and techniques that we have used as terminus a quo to our research and also as a benchmark to evaluate the usability of our methods and tools. During this research we review two of the most famous ontology learning tools Text2Onto[32] and OntoLT [18]as a delegate for the two common approaches, linguistic and statistical approaches. This section discusses how Text2Onto and OntoLT work and the difference that AOL has brought.

3.2. TEXT2ONTO

Text2Onto[32] is a framework for ontology learning from textual resources. It is a revision and advancement to their earlier framework TextToOnto [9] and it has three main features that they have improved on this version. First, the knowledge representation is made at a meta-level in the form of instantiated modeling primitives with what they called Probabilistic Ontology Model (POM).This lets the tool remain independent of a concrete target language (Ontology language) while being able to translate the instantiated primitives into any (reasonably expressive)knowledge representation formalism. Second, user interaction is also included as a main component in Text2Onto which is assumed to help calculate the confidence of the extracted ontology. Third, it has proposed a strategy for data-driven change discovery. In addition, they were able to avoid processing the whole corpus from scratch each time it changes; only selectively updating the POM according to the change in a corpus.

A Probabilistic Ontology Model (POM) as used by Text2Onto is a collection of instantiated modeling primitives which are independent of a concrete ontology representation language. Text2Onto includes a Modeling Primitive Library (MPL) which defines these primitives in a declarative fashion, which gives it two

advantages. On one hand, adding new primitives does not imply changing the underlying framework thus making it flexible and extensible. On the other hand, the instantiated primitives can be translated into any knowledge representation language. Thus, the POMs learned by Text2Onto can be translated into various ontology representation languages such as RDF[6], OWL and F-Logic[7].

The modeling primitives used in Text2Onto are:

- Concepts (CLASS)
- Concept inheritance (SUBCLASS-OF)
- Concept instantiation (INSTANCE-OF)
- Properties/relations (RELATION)
- Domain and range restrictions (DOMAIN/RANGE)
- Mereological relations (PART-OF)
- Equivalence (SIBLING)

The POM in Text2Onto is not probabilistic in a mathematical sense, because every instantiated modeling primitive gets assigned a value indicating how certain the algorithm in question is about the existence of the corresponding instance. The purpose of these 'probabilities' is to facilitate the user interaction by allowing him/her to filter the POM and thereby select only a number of relevant instances of the modeling primitives to be translated into a target ontology language.

In Text2onto data-driven change discovery is made to evolve the ontology as the source changes. Data-driven changes are generated by modifications to the underlying data, such as text documents or a database, representing the knowledge modeled by Ontology. Therefore, data-driven change discovery provides methods for automatic or semi-automatic adaption of Ontology according to modifications being applied to the underlying data set.

[6] RDF is a standard model for data interchange on the Web: http://www.w3.org/RDF/

[7] Frame Logic (or *F-logic*) provides a logical foundation for frame-based and object-oriented languages for data and knowledge representation: http://www.w3.org/2005/rules/wg/wiki/F-logic

3.3. ONTOLT

The OntoLT[18] approach follows similar three step procedure to ontology learning from text, 1) Concept extraction 2) taxonomic relations extraction and 3) non-taxonomic relations extraction. However it aims at directly connecting the ontology engineering with linguistic analysis through the use of mapping rules between linguistic structure and ontological knowledge. Linguistic knowledge (morphological and syntactic structure, etc.) remains associated with the constructed ontology and may be used subsequently in its application and maintenance.

OntoLT[18]takes a care of the ontology extraction process by first providing a precondition language, with which the user can define mapping rules. Preconditions are implemented as XPATH expressions over the XML-based linguistic annotation. If all constraints are satisfied, the mapping rule activates one or more operators that describe which way the ontology should be extended if a candidate is found. Predefined preconditions select for instance the predicate of a sentence, its linguistic subject or direct object. Preconditions can also be used to check certain conditions on these linguistic entities, for instance if the subject in a sentence corresponds to a particular lemma (the morphological stem of a word). The selected language will certainly consist of its distinct Terms and Functions.

Selected linguistic entities may be used in constructing or extending Ontology. For this purpose, OntoLT provides operators to create classes, slots and instances. According to which preconditions are satisfied, corresponding operators will be activated to create a set of candidate classes and slots that are to be validated by the user. Validated candidates are then integrated into a new or existing ontology.

Common to both mentioned frameworks is the use of some sort of natural language processing to derive features on the basis of which to learn ontological structures. And they also use an already existing and manually

crafted knowledge base, WordNet[51] in Text2Onto and EuroWordNet[50] in OntoLT.

OntoLT makes a number of the linguistic processing during the ontology extraction process and we can say it uses a linguistic approach. OntoLT needs the corpus to be highly annotated and pre-defined mapping rules are also expected from the user. Text2Onto on the other hand uses a machine learning approach to extract concepts and the relationship among them, and this is found to be computationally expensive which is demonstrated by the fact that it was running out of memory when run with ~250 Mbytes of text input on 64-bit computer reserving 3 GB of heap space.

In conclusion, Amharic ontology learner has a linguistic component that is uniquely designed to fit for Amharic language this shows that we cannot apply the above discussed frameworks for Amharic as they are. In addition to this AOL uses a hybrid approach which helps to take over the advantage from both approaches, statistical and linguistic. AOL is also not using any predefined knowledgebase like WordNet or EuroWordNet, which we believe is important for resource limited languages like Amharic, because such knowledge bases are not available for Amharic.

CHAPTER FOUR: AMHARIC ONTOLOGY LEARNER

4.1. INTRODUCTION

Amharic ontology learner is a learning system that automatically generates domain ontology from a plain text. To the best of the researcher's knowledge there is no research work conducted in the field of automatic concept extraction, relationship mining (among concepts) and formal representation of them on Ontology from Amharic text document.

Amharic ontology learner has the following two important features:

a) Produces ontology encompassing all possible concepts, both single word and multi-words.
b) Formalizes the resulting ontology using Web Ontology Language - OWL. In order to do so, we used Jena Semantic Web Framework. Jena provides a programmatic environment for RDF, RDFS and OWL and includes a rule-based inference engine. The use of OWL enables the visualization and inspection of the resulting Ontology using a common ontology editor such as Protégé.

4.2. OVERVIEW OF AMHARIC ONTOLOGY LEARNER

As discussed in Chapter 2, ontology learning involves the use of a set of interrelated activities. Amharic Ontology Learner works by accepting a set of Amharic documents focusing on a specific domain as input and generates an ontology that shows relevant and related concepts that can describe the domain.

The ontology learning process is shown graphically using the architecture presented in Figure 4.1. The architecture shows the different components involved in the ontology construction which are:

1. **Pre-processor:** This component is responsible to prepare the input corpus and make it suitable for ontology learning module. The pre-processing tasks include:
    - *Data cleaning*: Removing unwanted and unknown tags, removing un-recognized characters
    - *Sentence identification*: Separating every sentence and putting it in a separate line, so that the next module accesses every sentence separately.
    - *Part of speech (POS) identification:* Identifying the POS of every word from a tagged corpus.
2. **Concept extractor:** This module is dedicated to extract terms and map them into concepts when they are fit to. This research employs two methods/algorithms; TF-IDF for single word concepts extraction and C-Value for multi-word concepts extraction. These methods are discussed in detail in Section 4.4
3. **Taxonomy builder:** This component produces the taxonomy of concepts that are extracted in the previous step. Section 4.5 details the technique employed for this module.
4. **Non-taxonomic relations extractor:** This component enriches the taxonomy with any non-taxonomic relation that exists between concepts. The result of this component is very important in fully representing a given domain using ontology.

The input corpus we have used is a Parts of Speech Tagged multi-domain news document of WALTA information center. The corpus has passed through a pre-processing step, the annotations and the cleanings discussed above. The output from the pre-processing module is clean enough for any of the tools we have to apply to. Since ontology is constructed from a corpus of a certain domain, we classify the news into 7 known domains: namely Entertainment, Sport, Tourism, Business, Politics, Education and Social**.** The Tourism class is chosen to present our example and later on to test the performance of our

modules. The above four main modules of Amharic ontology learner will be discussed in the remaining sub-sections.

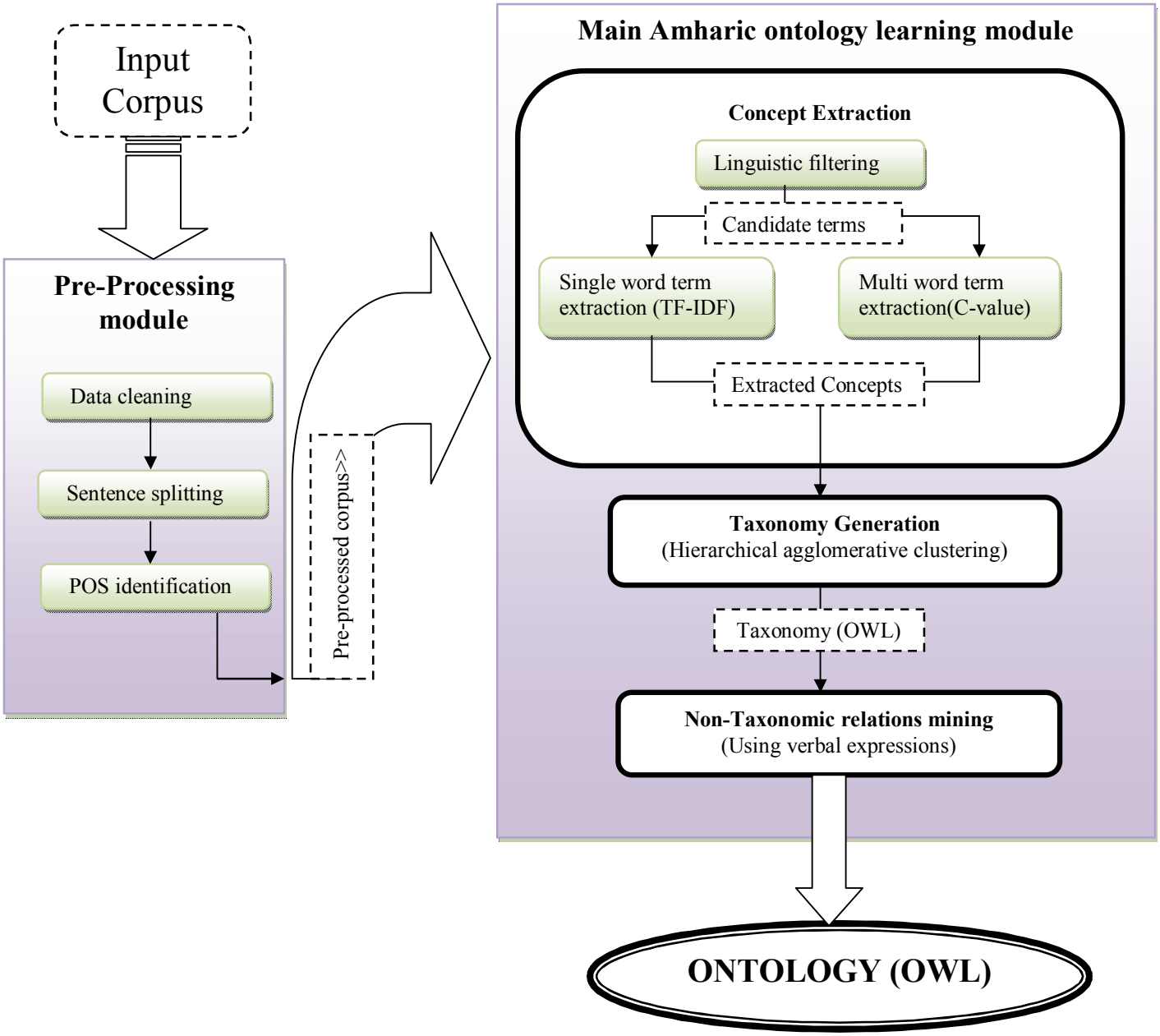

Figure 4.1: Amharic ontology learner architecture

4.3. PRE-PROCESSING

4.3.1. DATA CLEANING, SENTENCE SPLITTING & POS IDENTIFICATION

This module is dedicated to clean and structure the corpus and make it suitable for the upcoming modules to process. The input corpus is found to contain a number of unwanted tags, characters, etc. For instance, the corpus contains XML declaration statement, such as <?xml version="1.0" ?> and<! DOCTYPE amnews94>. These and other unwanted tags are cleared out as they are not important to this research.

In Natural Language Processing, a sentence is a string of words satisfying the grammatical rules of the language and talks about one core topic. In this research, sentences are considered as basic unit for manipulation and hence are identified separately and manipulated using our sentence splitter sub-component. Refer to Appendix A for its implementation

In ontology learning the word class or Parts Of Speech of a word is very important and it is used in linguistic filtering and non-taxonomic relations mining. For this purpose we have implemented a tool that could identify the POS of a word from an already tagged corpus. Refer to Appendix B to see its implementation for linguistic filtering.

4.3.2. LINGUISTIC FILTERING

A linguistic filtering is a process by which words/phrases satisfying a certain linguistic criteria are taken in or words/phrases not satisfying a certain linguistic criteria are taken out. The criterion can be the word class of a single word or a word class sequence in sequence of words. E.g. A filtering criteria 'N' will let any Noun, even a Noun surrounded by other words from different word classes to be passed but a filtering criteria 'N+' will let only Noun sequences, more than one nouns successively appearing in a corpus, to pass.

A linguistic filtering technique is one of the techniques that can be used for term extraction, but it is better to use as a facilitator or as an enthusiast for other term extraction methods than as a term extractor itself because terms occur in different patterns and may also be in complex sentences that we cannot set linguistic rules. In this research we used a linguistic filter to remove stop words and words that can hardly be a term or part of a term. Stop words are removed because their higher frequency in a document may lead to a wrong conclusion. In this research, we used the stop words list presented in Appendix C, and the stop word remover that is developed by Tessema Mindaye et al [5].

The linguistic filter is also employed to remove words from word classes that are suggested by a linguist to hardly constitute a term. This linguistic filtering can be applied in two ways, closed filter and open filter. Closed filter is the one in which only those that firmly satisfies the linguistic criteria are passed, whereas open filter can be considered as a somehow relaxed filtering technique.

The choice of linguistic filter affects the precision and recall of the output list. A closed filter which is strict about the strings it permits will have a positive effect on precision but a negative effect on recall and vice versa [44]. As an example, consider the "N+" filter that Dagan, Church used [44]. This filter permits only noun sequences and as a result produces high precision since noun sequences in a corpus are the most likely to be terms. At the same time, it negatively affects recall, since there are many noun compound terms that consist of adjectives and nouns, which are excluded by this filter.

We chose to make an open filter that prevents only word classes known to hardly constitute a concept, which results in augmented recall over precision: The word classes suggested by a linguist to be filtered out are 'PREP', 'PUNC' , 'PRONP' , 'AUX' , 'PRON' , 'V' , 'VP' , 'VC' ,'VREL so we filtered them out.

Once the stop lists and un-wanted word classes are removed from the corpus, words having a better probability of being a term/part of a term will be left.

4.4. CONCEPT EXTRACTION

A concept is a mental symbol, which represents class of things denoting similar fact. This Section presents a technique employed for automatic extraction of concepts. Automatically extracting concepts is very important, as it saves time, computation and man power, but it is also very difficult, due to the following gaps.

The first of the limitations is that we cannot represent every concept with only one word, as there are concepts composed of several words such as ባህል ማስታወቂያና ቱሪዝም ቢሮ and ዋልታ የኢንፎርሜሽን ማእከል which needs to be recognized too. This research handles this limitation by employing two different concept extraction algorithms, one for a single-word concept extraction (TF-IDF) and another for a multi-word concept extraction (C-Value).

The second of the limitations is that, different lexicons may be extracted as a separate concept but found to implicitly represent the same concept, these needs to be normalized to get a term to a concept correspondence, which is important to finally obtain a neat knowledge. The analysis on those variants showed that they mostly occur due to morphological variations. As a morphologically rich language, Amharic suffers from this multiplicity, caused by different morphological variants of a single word; we tackle this by clustering concepts into one when their lemmatized form is the same. This is implemented differently for single word and multi-word terms.

For single word terms we took the lemmatized form of all terms and clustered into one when there are terms having similar lemma. E.g. ‘ፓርክን’ ‘በፓርኩ’, ‘ፓርኩ’ and ‘ፓርክ’ all represent the same concept ‘ፓርክ’. So we cluster all those term variants into one by considering their root form (ፓርክ).The concepts are also presented in ontology in their root form so that future existence of the same concept in a different form is recognized.

For multi-word terms a similar procedure is followed except the fact that the lemmatization is done only to the head (the first word from a multi-word term).This is because the morphological variation to a multi-word term mostly occurs at its head. So, all multi word terms having the same lemmatized head are grouped and the comparison is made on their modifiers (words other than the first word in a term), if any of these terms have similar modifiers we cluster them into one as they are representing the same concept.

For instance, the term ለዋልታ ኢንፎርሜሽን ማእከል ,የዋልታ ኢንፎርሜሽን ማእከል, ከዋልታ ኢንፎርሜሽን ማእከል and ዋልታ ኢንፎርሜሽን ማእከል are extracted as separate terms. The normalization is began by checking if those different terms have the same lemmatized head (ዋልት) from their original heads (ለዋልታ, ከዋልታ, የዋልታ, ዋልታ). In addition, all these terms have similar modifiers, ኢንፎርሜሽን ማእከል, thus, these terms are understood to represent the same fact and inherently represent the concept ዋልት ኢንፎርሜሽን ማእከል.

Our choosing of TF-IDF (for single word concepts) and C-Value (for multi-word concepts), is following the result from a comparative evaluation of existing automatic term recognition techniques by Ziqi Zhang [11], Table 4.1.

Table 4.1: Comparative evaluation of popular term recognition algorithms on GENIA corpus.

Total Number of terms evaluated = N					
N	**TF-IDF**	**WEIRDNESS**	**C-VALUE**	**GLOSSEX**	**TERMEX**
100	0.9	0.48	0.91	1	0.92
1K	0.82	0.55	0.91	0.82	0.75
5K	0.8	0.58	0.83	0.69	0.62
10K	0.75	0.58	0.68	0.66	0.61
20K	0.6	0.56	0.58	0.59	0.55

4.4.1. TF-IDF IMPLEMENTATION

TF-IDF algorithm is used in Amharic ontology learner to extract single word concepts, It is to be recalled from Section 4.2thatthe heterogeneous news corpus, used as input, is classified into 7 predefined categories (7 different domains), and these seven domain documents are used as a document collection to apply TF-IDF. Calculating the TF (Term frequency) of a term **T**in a certain document **D** is straight forward; find the raw count of a **T** in **D**, but to calculate the IDF for a term in a document we need to have a reference corpus. In this research the IDF of a term in one document is calculated using Equation 2.1, considering the other documents in entire WIC News corpus as a reference. i.e., to calculate the IDF for a term in a tourism document, we will keep the Sport, Entertainment, Politics and other domain documents as a reference corpus.

When this module is run a total of 3717 single word concepts, with TF-IDF value greater than 0 are extracted. For ease of presentation and manageability we have presented the top 50 candidate concepts in Table 4.2.The top 50 candidate concepts are normalized, as detailed above in Section 4.4, and a result presented in Table 4.3 is achieved**.**

Table 4.2: List of candidate single word concepts from a TF-IDF module before normalization

Rank	TFIDF	Concept
1	0.593414	የዱር
2	0.257718	የቱሪዝም
3	0.256041	ሱባ
4	0.235628	እንስሳት
5	0.214288	ሙዚየም
6	0.204833	ውቅር
7	0.204833	እላት
8	0.200676	ቱሪስቶች
9	0.200676	መስህቦች
10	0.184084	ቅርሶች
11	0.179229	በፖርኩ
12	0.179229	በመናገሻ
13	0.179229	ቅሬተ

14	0.156081	በፓርኩ
15	0.156081	ቅርሶችን
16	0.153624	ታቦት
17	0.148354	አጥቢ
18	0.148354	አቡነ
19	0.148354	ነብር
20	0.148354	ተራሮች
21	0.148354	ቅርስ
22	0.133784	ፓርክ
23	0.13187	ቢኖሩም
24	0.13187	ቆርኬ
25	0.128391	ሃላፊ
26	0.12802	የስዌን
27	0.12802	አእዋፍ
28	0.12802	በስኮትላንድ
29	0.123963	ክልል
30	0.115386	ጎብኚዎች
31	0.115386	የላሊበላ
32	0.115386	ቀበሮዎች
33	0.111487	ዝርያዎች
34	0.111487	ኮሚሽነሩ
35	0.11045	መግለጫው
36	0.103087	ደን
37	0.103087	ባህላዊ
38	0.102416	ፖርክን
39	0.102416	ፓጳስ
40	0.102416	ገሪማ
41	0.102416	የነብር
42	0.102416	የቱሪስቶች
43	0.102416	የተራራ
44	0.102416	እንዲመለስ
45	0.102416	እላቱን
46	0.102416	እላቱ
47	0.102416	ኢሬክተስ
48	0.101827	ሃላፊው
49	0.100338	ባለሙያው
50	0.098902	ፓርኩን

Table 4.3: List of candidate single word concepts from a TF-IDF module after normalization

Rank	TF-IDF	Concept
1	0.431565	ዱር
2	0.42282	ፓርክ
3	0.415888	ፖርክ
4	0.347803	ቅርስ

5	0.339642	ስሳት
6	0.311916	ቆርክ
7	0.286702	ሃላፍ
8	0.277259	ዝርይ
9	0.277259	ሙዚይ
10	0.270327	አእዋፍ
11	0.263396	ተራር
12	0.207944	አጥብ
13	0.207944	ስኮትላንድ
14	0.207944	ሱብ
15	0.194081	ኑበር
16	0.192701	ደን
17	0.18715	አቡን
18	0.18715	ታቦት
19	0.18715	መናገሸ
20	0.180218	መቅደል
21	0.180218	ላሊበል
22	0.176549	ጎበኝ
23	0.173901	ባለሙይ
24	0.156933	እድም
25	0.156933	ቀበር
26	0.155101	አካባብ
27	0.155101	ተፈጥር
28	0.145561	አእዋፋት
29	0.140964	ቢር
30	0.132334	ቱሪስት
31	0.124766	ድኩል
32	0.124766	ደንቆር
33	0.124766	ተፈጥሮአው
34	0.124766	ስው
35	0.124766	ርል
36	0.120178	ቱሪዝ
37	0.117835	ትበቅ
38	0.117501	መምሪይ
39	0.110904	ገዳም
40	0.109496	ክልል
41	0.106442	ገለጹት
42	0.103972	ዝንጀር
43	0.103972	ዝሀ
44	0.103972	ቤተክርስቲይ
45	0.103972	ቁም
46	0.103972	ሰፋር
47	0.103972	ሙዚየም
48	0.103972	ሆም
49	0.097812	መስህብ
50	0.097041	ዩኔስክ

The above two tables show that the normalized result in Table 4.3 is better as compared to the results in Table 4.2.This is demonstrated by the fact that we have 5(five) variants of the term ፓርቲ in the original result (Table 4.2: 11th, 14th, 22nd, 38th and 50th), whereas we only have only 2 variants of the same term ፓርቲ in the normalized list (Table 4.3: 2nd and 3rd). This tells us that normalizing the single word terms can give us a better result.

4.4.2. C-VALUE IMPLEMENTATION

Amharic Ontology learner uses C-value method to extract multi-word concepts, and the algorithm we proposed for using C-value method in AOL is shown in Algorithm 4.1.

The C-value Algorithm (Written as applied to AOL)

Input: All candidate terms

Line

1. For all strings **a** of maximum length
2. Calculate$C-calue(\boldsymbol{a}) = \log_2|\boldsymbol{a}|.f(\boldsymbol{a})$;*// for terms that are not found as a sub-string to another terms.*
3. if C-value(**a**) >= Threshold
4. add **a** to output list;
5. For all substrings **b***// For terms that are found nested in another term.*
6. revise t(**b**);*// Calculates the frequency of b as a sub-string of other terms.*
7. revise c(**b**);*// Calculates the number of those longer candidate terms.*
8. For all smaller strings **a** in descending order
9. if **a** appears for the first time
10. $$C-calue(a) = \log_2|a|.f(a)$$
11. Else
12. $$\mathrm{C-calue(a)} = \log_2|\mathrm{a}|.\mathrm{f(a)} - \frac{1}{\mathrm{P(Ta)}}\sum_{\mathrm{b}\epsilon\mathrm{Ta}}\mathrm{f(b)}$$
13. if C-value(**a**) >= Threshold

14. add **a** to output list;
15. for all substrings **b**
16. revise t(**b**);*// Calculates the frequency of b as a sub-string of other terms.*
17. revise c(**b**);*// Calculates the number of those longer candidate terms.*
18. Gets the output; *// List of C-values for all candidate terms*

Algorithm 4.1: Multi-word concepts extracting algorithm

The C-value algorithm is implemented in two steps:

Step 1 (Candidate terms extraction)

In this step we generate all possible combinations of word sequences in every sentence (bigram, trigram … ngram); which are considered as candidate terms.
E.g. Given Sentence: = {በደንቆሮ ደን የቀይ ቀበሮ ቁጥር}
Candidate terms: = {በደንቆሮ ደን}, {በደንቆሮ ደን የቀይ}, {በደንቆሮ ደን የቀይ ቀበሮ} and
{በደንቆሮ ደን የቀይ ቀበሮ ቁጥር},{ደን የቀይ},{ደን የቀይ ቀበሮ},{ደን የቀይ ቀበሮ ቁጥር},{የቀይ ቀበሮ},
{የቀይ ቀበሮ ቁጥር},{ቀበሮ ቁጥር}.

As it is shown in the above example the term length (word count) can go as large as the length of the sentence. According to [45] the maximum length of the extracted strings may depend on:

1. **The domain of the text in consideration**. In arts for example, terms tend to be shorter than in science and technology.
2. **The nature of the term**. E.g. Terms that only consist of nouns very rarely contain more than 5 or 6 words.

The input corpus used in this research is neither art nor science, but news, and we are not only selecting noun compounds as candidate terms. This keeps us free from the previous restriction; thus we allow strings up to the whole sentence to be considered. Having all these possible candidates will surely result in a much number of candidate terms; which seems to imply that we

have to calculate the C-value for all of them, but this isn't happening because candidates which occur in a corpus less than the specified threshold are filtered out. We tried the threshold at different numbers, 2,3,4,5 and 6 and when it is set to 4 a better result is achieved. It takes much time and memory to calculate the C-value for longer candidate terms. However longer and rare candidate terms will not create computational overhead as they won't have enough frequency in the corpus. By the end of this step we have list of candidate terms that occur above the frequency threshold, which is four, and we go for computing their C-values,

Step 2: (Computing C-value)

Once the candidate terms are defined, we used Algorithm 3.1 and the techniques detailed in Section 2.4 to calculate the C-value measure of terms.

When the C-value computing module is run it gives us 75 multi-word terms with C-value greater than 0 and the top 50 are shortlisted in Table 4.4. Taking the assumption discussed in Section 4.4 the multi-word terms were also normalized, but no significant improvement was noticed.

Table 4.4: List of candidate multi word concepts using a C-value module before normalization

Rank	Concept	C-Value
1	እንፎርሜሽን ማእከል	19.5
2	የዱር እንስሳት	17.33333
3	ለዋልታ እንፎርሜሽን ማእከል	11.69925
4	ዋልታ እንፎርሜሽን ማእከል	10.61429
5	ባህል ቱሪዝምና ማስታወቂያ	4.509775
6	ተራሮች ብሄራዊ ፓርክ	4.094738
7	ባህል ማስታወቂያና ቱሪዝም ቢሮ	4
8	ባህል ቱሪዝምና ማስታወቂያ ቢሮ	4
9	የዞኑ ንግድ እንዱስትሪና ቱሪዝም መምሪያ	3.965784
10	ተራሮች ብሄራዊ	3.5
11	ባህል ማስታወቂያና ቱሪዝም	3.424813
12	ንግድ እንዱስትሪና ቱሪዝም	3.08985
13	የዞኑ ንግድ እንዱስትሪና ቱሪዝም	3
14	ንግድ እንዱስትሪና ቱሪዝም መምሪያ	3
15	የዱር እንስሳት ከፍተኛ ባለሙያ	3

16	ብሄራዊ ፓርክ	3
17	በቢሮው የባህል መምሪያ ሃላፊ	3
18	የቅርስ ትናትና ትበቃ ባለስልጣን	3
19	የኢትዮጵያ ቱሪዝም ኮሚሽን	2.924813
20	ለዋልታ ኢንፎርሜሽን ማእከል	2.924813
21	የቅርስ ትናትና ትበቃ	2.83985
22	ቱሪዝም ኮሚሽን	2.666667
23	መምሪያ ሃላፊ	2.666667
24	ኢንፎርሜሽን ማእከል	2.666667
25	የአለም ቱሪዝም ቀን	2.33985
26	የጽህፈት ቤቱ ሃላፊ	2.33985
27	ውቅር አብያተ ክርስቲያናት	2.33985
28	ማስታወቂያና ቱሪዝም ቢሮ	2.33985
29	ቱሪዝምና ማስታወቂያ ቢሮ	2.33985
30	የቀይ ቀበሮዎች ቁጥር	2.33985
31	የዞኑ ንግድ እንዱስትሪና	1.754888
32	እንዱስትሪና ቱሪዝም መምሪያ	1.754888
33	ቅዬ የዱር እንስሳት	1.754888
34	የዱር እንስሳት ከፍተኛ	1.754888
35	እንስሳት ከፍተኛ ባለሙያ	1.754888
36	የሱፍ አብዱላሂ ሱከር	1.754888
37	የዱር እንስሳት እና	1.754888
38	ቅርሶችን የመጎብኘት ባህሉ	1.754888
39	የአዲስ አበባ ሙዚየም	1.754888
40	የሰሜን ተራሮች ብሄራዊ	1.754888
41	በቢሮው የባህል መምሪያ	1.754888
42	የባህል መምሪ ያሃላፊ	1.754888
43	እንፎርሜሽን ማእከል መግለጫ	1.754888
44	በመቅደላ ቶርነት ወቅት	1.754888
45	ትናትና ትበቃ ባለስልጣን	1.754888
46	የባሌ ተራሮች ብሄራዊ	1.754888
47	የብጹእ ወቅዱስ ፓትሪያርክ	1.754888
48	የቱሪዝም አማካሪ ምክር	1.754888
49	የድንጋይ ዘመን ሰዎች	1.754888
50	በቤንሻንጉል ጉሙዝ ክልል	1.754888

Once the concept extraction module have recognized all possible terms the next step is to discover all possible relationships that exist between concepts. This relationship mining process is simplified into two different steps: First, all possible taxonomic(Hierarchical/is a) relationships are extracted and second any non-taxonomic relations are mined. The methods employed for each of the above two tasks are discussed in the following sections:

4.5. TAXONOMIC RELATIONS MINING

Once relevant concepts are extracted using the techniques presented in previous sections there is a need to identify the relationship between them. This particular section discusses the method employed to automatically generate taxonomy.

There are different methods for automatically building taxonomy of concepts and most of them uses a pre-defined knowledge base like wordnet[17, 32].Since this research is for Amharic and there is no 'Amharic wordnet' we chose to use a method that does not use a pre-defined knowledge base. Hierarchical agglomerative clustering [47] is a technique we chose for taxonomy building module, mainly due to its independence on external knowledge,

This backbone hierarchy or the taxonomic tree will be transformed into a full relationship grid later after it is enriched with non-taxonomic relations. Implementation of the taxonomic relationship mining techniques is discussed next.

4.5.1. HIERARCHICAL AGGLOMERATIVE CLUSTERING

Hierarchical agglomerative clustering is a kind of clustering which proceeds bottom-up; the largest cluster is formed at the end. In this research the clustering starts with every concept as individual cluster and at each step it computes the similarity between all pairs of clusters and merges the most similar pair.

Though the algorithm typically continues until a single cluster is formed at the top of the hierarchy, in this research it stops when there is no similar constituent word among concepts to be related. We used the group average method to compute the similarity between two clusters. Being more specific, the group average method computes the average similarity across all pair of concepts within the two clusters (Ci,Cj) [47]:

$$sim(Ci, Cj) = \frac{\sum_{x \epsilon ci, y \epsilon cj} \text{Sim(x,y)}}{|\text{ci}| * |\text{Cj}|} \quad \text{..... 4.1}$$

Where

- x is a concept in cluster Ci and
- y is a concept in cluster Cj correspondingly

We proposed algorithm 4.2 to apply Hierarchical agglomerative clustering for taxonomy building in AOL.

1. Load all concepts.
2. Treat each concept as a cluster on its own.
3. Compute the similarity between all pairs of clusters; calculate the similarity matrix whose i^{th} entry gives the similarity between the i^{th} and j^{th} clusters.
4. Merge the most similar two clusters using the group average similarity.
5. Update the similarity matrix entries for the newly formed cluster and the other clusters.
6. Repeat steps 4 and 5 until desired clustering level is reached.

Algorithm 4.2: Taxonomy building algorithm

The time complexity of a typical hierarchical agglomerative clustering algorithm is O (n^2) where n is the number of concepts.

In this research Lexical similarity is used to measure similarity between extracted concepts. This idea was first proposed by Bourigault and Jacquemin[12] by considering the heads(The first word/words in a concept) of two concepts. This approach is generalized by considering constituents (head and modifiers) shared by concepts. Most of the time a concept gets children when any other word, mostly an adjective is attached to the parent concept:

e.g. The Concept እንስሳት/Animals get a child የዱር እንስሳት/Wild animals when the adjective የዱር/Wild is attached to it. In Amharic Adjective comes in front, so the parent word stays at the rear of a multi-word term. This research uses the technique proposed by Bourigault & Jacquemin[12] in a way that makes it applicable for Amharic.

The rationale behind lexical similarity involves the following hypotheses:

A. Terms sharing a rear are assumed to be direct hyponyms of the same term, the rear. e.g. የዱር እንስሳት(Wild Animals) and የቤት እንስሳት(Domestic Animals) are both እንስሳት(Animals).
B. When a term is nested inside another term, we assume that the terms are related: e.g. የሰሜን ተራሮች ብሄራዊ ፓርክ (North Mountains National park) and ብሄራዊ ፓርክ (National park).When this case happens the relation is created in a way that the term included in another larger term is an upper class to the longer candidate term. ብሄራዊ ፓርክ (National park) is an upper concept in hierarchy to የሰሜን ተራሮች ብሄራዊ ፓርክ (North Mountains National park).

The lexical similarity between terms t_1 and term t_2 (whose heads are denoted by h_1 and h_2 respectively) is computed according to equation 4.2.

$$LS(t1,t2) = \frac{|P(h1)\cap P(h2)|}{|P(h1)+P(h2)|} + \frac{|P(t1)\cap p(t2)|}{|P(t1)+P(t2)|} \quad \text{..... 4.2}$$

Where: P(h1): is number of combinations that a head of term 1 can make.

P(t1): is the number of combinations that term 1 can form.

P(h2): is number of combinations that a head of term 2 can make.

P(t2): is the number of combinations that term 2 can form.

The numerators in equation 4.2 denote the number of shared constituents, while the denominators denote sums of total numbers of constituents. Some sample terms and their lexical similarity value is shown in Table 4.5

The taxonomy building process continues as long as there are terms sharing a rear word. Portion of the taxonomy (is-a hierarchy) generated using hierarchical agglomerative clustering technique is shown in Figure 4.3and the directed graph representation of the same result can be viewed on protégé in Figure 4.2.

Table 4.5: Lexical similarity Example

i	t_i	$P(t_i)$
1	ፓርክ	{ፓርክ}
2	ብሄራዊ ፓርክ	{ብሄራዊ}, {ፓርክ}, {ብሄራዊፓርክ}
3	የሰሜን ተራሮች ብሄራዊ ፓርክ	{የሰሜን}, {ተራሮች}, {ብሄራዊ}, {ፓርክ}, {የሰሜን ተራሮች}, {ተራሮች ብሄራዊ}, {ብሄራዊ ፓርክ}, {የሰሜን ተራሮች ብሄራዊ}, {የሰሜን ተራሮች ብሄራዊ ፓርክ}
4	ኦሞ ብሄራዊ ፓርክ	{ኦሞ}, {ብሄራዊ}, {ፓርክ}, {ኦሞ ብሄራዊ}, {ብሄራዊ ፓርክ}, {ኦሞ ብሄራዊ ፓርክ}

The Lexical similarity matrix for T1,T2,T3 and T4

	T_1 - ፓርክ	T_2 - ብሄራዊ ፓርክ	T_3 - የሰሜን ተራሮች ብሄራዊ ፓርክ	T_4 - ኦሞ ብሄራዊ ፓርክ
T_1 - ፓርክ	1.00	0.75	0.60	0.64
T_2 - ብሄራዊፓርክ	0.75	1.00	0.75	0.83
T_3 - የሰሜን ተራሮች ብሄራዊ ፓርክ	0.60	0.75	1.00	0.70
T_4 - ኦሞ ብሄራዊ ፓርክ	0.64	0.83	0.70	1.00

Once all concepts and their taxonomic relations are defined there is a need to enrich the acquired taxonomic tree with domain specific and any other possible relation that could exist among them, so that the taxonomy is grown into a full ontology. The next section discusses the implementation of the non-taxonomic relations extractor module.

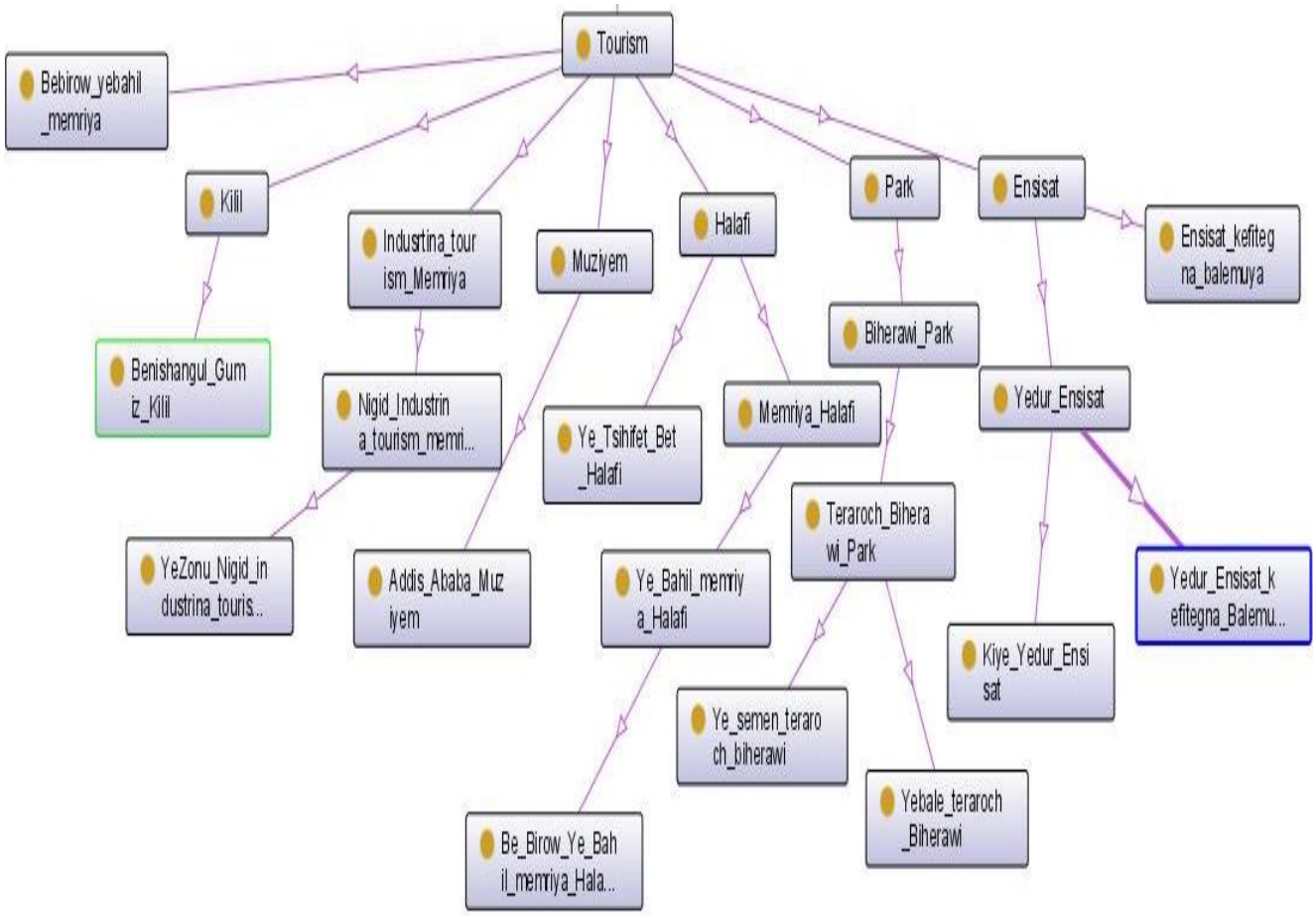

Figure 4.2: Graph representation of the taxonomy

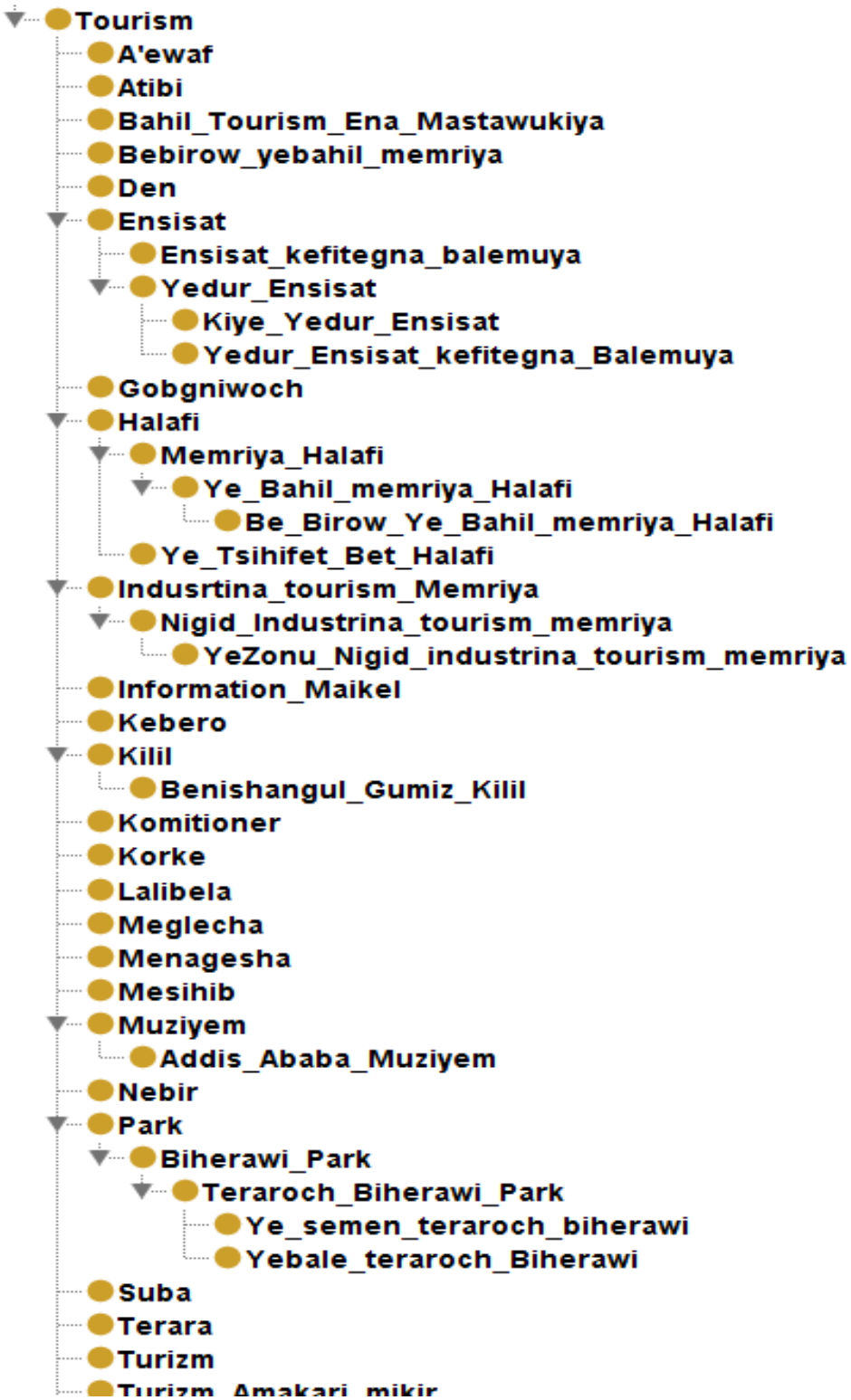

Figure 4.3: Sample domain taxonomy from tourism corpus

4.6. NON-TAXONOMIC RELATIONS

These include all possible relations that could exist between concepts in a certain corpus other than a hierarchical 'is a' relations. Non-taxonomic relations are important in growing our taxonomy into ontology. Non-taxonomic relations are difficult to recognize from a corpus as they come in dynamic pattern. Thus, it is difficult to use a rule-based or any statistical method to extract them. This research employs a linguistic method, using verbal expressions as a relation indicator, to discover non-taxonomic relations.

The Amharic ontology learner extracts non-taxonomic relations in the following manner; Concepts in the same sentence are thought to be related and a relation among them is described by the verb in that sentence. This module will parse every sentence and tries to extract all possible relations that could exist among concepts in that sentence. The technical detail is stated below.

4.6.1. USING VERBAL EXPRESSIONS AS A RELATION INDICATOR

The formal grammatical structure of a regular simple Amharic sentence is Subject (ድርጊት ፈጻሚ) + Object (ድርጊት ተቀባይ) +Verb (ግስ). The verb in a sentence describes the relation between the subject and the object of that sentence. In this research we consider the verb as an indicator to the relation between the subject and object of the sentence. But in the corpus that we have worked on most of the sentences are complex, sentences with more than one noun phrase and verb phrase. This means, there may be more than two concepts in a sentence. Thus, it is not easy to find which two of the concepts in a sentence are related with which of the verbs, so the sentence needs to be analyzed. The steps followed in implementing this module are:

First: Consider two concepts in a sentence.

Second: Consider all verbs that occur after those two concepts in that sentence. Verbs at the end are considered because the verb at the end is the one belonging to the noun of the first noun phrase [19,20].

Third: Choose the verb that related those concepts.

In a sentence the subject is the one doing the action/verb on an object and in Amharic a relation between the subject and an object is represented by the verb that comes at the end of the sentence. In complex Amharic sentences, there are a number of verbs, and a refinement is needed to determine which verb relates which two concepts.

In this research, the first concept to occur in a sentence is considered as the domain and all concepts that are found next in the same sentence are considered as the range to the first concept. In every pair of those concepts the verb to be considered as a relation indicator between them is the one among those verbs that comes after both concepts. After a number of trials we found that the verb that comes at the end is the one that better represents the relation between concepts in that sentence. Since the input corpus is a news in most of the sentences, the final words were found to be ዘግቧል, አስታወቁ, ገለጹ and ዘገበ … so a check is made if the final word is not among the above stated ones but if it is, we will consider the second from the last.

This way the relations mining process continues for every sentence and all possible relations that are there in a corpus are recognized, but all these relations will not be necessary to represent a domain in ontology. The concepts in a relationship should be represented at their appropriate level of generalization. E.g. the concepts ሰፋሪዎች and ህገ-ወጥ ሰፋሪዎች appear as a domain to the relation መጨፍጨፍ indifferent sentences, similarly the concepts ደን, የደንቆሮ ደን and የተፈጥሮ ደን appear as a range with the same relation in different sentences.

This will result in multiple relations like the ones shown in Table 4.6, which needs to be generalized up to the level appropriate in representing a domain in question. If we consider the previous example the generalized relation that needs to be represented in ontology is መጨፍጨፍ (ህገ-ወጥ ሰፋሪዎች,ደን).

There are methods for finding this proper generalization level of concepts; According to Cimiano[27], the conditional probability outperforms the others, thus we applied the conditional probability measure to detect the most appropriate generalization for the concept in question. The conditional probability method works as follows:

Taking the previous example the domain concepts ሰፋሪዎች and ህገ-ወጥ ሰፋሪዎች appears 22 times each, and the concepts ደን, የደንቆሮ ደን and የተፈጥሮ ደን appear 13,10 and 8 times respectively. Once the raw count of the concepts as a domain and range of a relation is found their respective conditional probability is calculated and the one with higher conditional probability is taken, if two or more concepts have the same conditional probability the most specific one according to the domain taxonomy is taken.

Table 4.6: List of relations to find the appropriate level of generalization

Domain	**Range**	**Relation**
ሰፋሪዎች	ደን	መጨፍጨፍ
ሰፋሪዎች	የደንቆሮ ደን	መጨፍጨፍ
ሰፋሪዎች	የተፈጥሮ ደን	መጨፍጨፍ
ህገ-ወጥ ሰፋሪዎች	ደን	መጨፍጨፍ
ህገ-ወጥ ሰፋሪዎች	የደንቆሮ ደን	መጨፍጨፍ
ህገ-ወጥ ሰፋሪዎች	የተፈጥሮ ደን	መጨፍጨፍ

Thus ሰፋሪዎች has a conditional probability of22/44 = 0.5 and ህገ-ወጥ ሰፋሪዎች has the conditional probability of22/44 = 0.5, we found that these two concepts have the same conditional probability 0.5, so ህገ-ወጥ ሰፋሪዎች is taken as it is the most specific one in the taxonomy, see Figure 4.4.Similarly, the conditional probability for the concepts as a range became 13/31 = 0.42 for ደን 10/31 = 0.32 for የደንቆሮ ደን and 8/31 = 0.26 for የተፈጥሮ ደን, so we kept ደን to be the appropriate level of generalization as it has the highest conditional probability.

This gives us the appropriate relation መጨፍጨፍ (ህገ-ወጥ ሰፋሪዎች, ደን). Portion of the non-taxonomic relations are shown in Table 4.7.

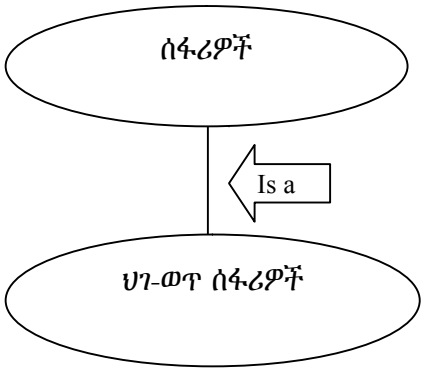

Figure 4.4: A taxonomic tree for ሰፋሪዎች concept

The objective of Amharic Ontology learner is to generate ontology from unstructured text input, so once we have all concepts, the taxonomy tree and other non-taxonomic relations the final step is representing the ontology using OWL. Jena semantic web framework and protégé are used to generate and visualize the ontology in a way presented in Figure 4.2 and Figure 4.3. The concepts and relations are mapped to OWL according to the following correspondence.

- Concepts ⟹ Classes
- Taxonomy ⟹ Sub-class-of
- Non-taxonomic relations ⟹ Object Properties/labeled by the verb

Portion of the generated code during ontology construction on protégé can be found in Appendix E.

Table 4.7: Sample non taxonomic relations

Non-Taxonomic Relations		
Domain	Range	Relationship/ Label
ቱሪዝም	ውቅር	መነብኘት
ቱሪስት	ቱሪዝም	መነብኘት
ቱሪዝም	መስህቦች	መነብኘት
ቱሪዝም	ቅርሶች	መጠበቅ
ፓርክ	ቱሪዝም	እንደሚዘጋጅ
ተራቶች	ቱሪዝም	--------
ሃላፊ	ቱሪዝም	አስረድተዋል
ክልል	ቱሪዝም	አለመካሄድ
ጎብኚዎች	ቱሪዝም	ጎብኝተዋል
ቱሪዝም	ላሊበላ	--------
ቱሪዝም	ኮሚሽነር	እንደሚፈታ
ቱሪዝም	መግለጫ	--------
የአለም የቱሪዝም ቀን	ቱሪዝም	መሳተፍ
ሱባ	እንስሳት	እንደሚገኙ
ሱባ	መስህቦች	እንደማይተናነስ
መናገሻ	ሱባ	ተካሂዷል
ኑብር	ሱባ	እንደሚገኝ
አእዋፍ	ሱባ	የቻሉት
ሱባ	ደን	---------
ሱባ	ተራራ	መውጣት
ሱባ	ሃላፊው	እንዲመደቡለት
ሱባ	የዱር እንስሳት	እንደሚገኝ
ቱሪስቶች	እንስሳት	ተጠናቋል
ፓርኩ	እንስሳት	እንደሚገኙ
እንስሳት	አጥቢ	የሚገኙ
ኑብር	እንስሳት	እንደሚገኙ
ቆርቄ	እንስሳት	መበራከት
ደን	ቀበሮዎች	እንደነበሩ

CHAPTER FIVE: EVALUATION AND RESULTS

5.1. INTRODUCTION

This chapter discusses how the evaluation of modules is done in Amharic ontology learner. Evaluating an ontology learner is difficult task because of three main reasons; first, there is neither concept extractor nor relationship miner that we can use as a benchmark to compare our system with. Second, manually extracting concepts and relations is not applicable to a large corpus and third, even if the second case was manageable there is no standard way to model a domain in question.

In this research, the evaluation is done on a portion of the output through manual inspection. The top 50 results from single-word term extractor, the top 50 results from a multi-word term extractor and the relations that are found to exist among those 100 terms will be manually inspected by linguists. The result from the linguist's feedback will be summarized using statistical measure, a precision.

Precision and recall are known to be the best statistical measures for methods and tools in Information retrieval. Precision is the fraction of retrieved instances that are relevant, whereas recall is the fraction of relevant instances that are retrieved. Recall is not applicable in this research as it is difficult to find all true concepts and their relations with which our result can be compared to. So, the same as other similar researches [2, 4, 18, 32] precision is a measure used in this research.

To the knowledge of the researcher, there is no previous research made regarding Amharic ontology learning with which a comparison can be made.

The performance of modules in ontology learning system depends on different factors, including but not limited to:

- ***The corpus used:*** This research is tested with a news corpus which is not the best for term extraction and relationship mining as compared to a corpus with higher frequency of domain terms.
- ***The performance of NLP tools used:*** The performance of the NLP tools used such as stemmer, tagger and parser determines the performance of the ontology learning system as a whole and
- **The nature of the language itself*:*** Amharic is one of the morphologically rich languages, which makes it difficult to the NLP tools used.

As it is stated above we have made the evaluation through manual inspection of portion of the results. Our input corpus is a heterogeneous multiple domain news corpus from WALTA information center. Before the testing is conducted, we classified the whole corpus into 7 classes each assumed to represent a different domain; these are Sport, Education, Entertainment, Politics, business, Tourism and Social news. We randomly chose the corpus that is classified as tourism to test every module of our Amharic ontology learner.

The performance of every module is quantified using precision. We took the top 50 results from the single word concept extractor and multi-word concept extractor modules each. The relationship miner modules (taxonomic relations miner and non-taxonomy relations miner) are evaluated based on the relations they were able to extract among those 100 (50 single word and 50 multi-word concepts) concepts.

The testing corpus, a news corpus from a tourism domain, is written in a way that anyone, not necessarily someone from that specific domain can understand, so we chose a linguist to judge our results.

The evaluation steps and the results found for each module will be discussed in subsequent sections.

5.2. CONCEPT EXTRACTOR

The extraction of concepts is a prerequisite for all main tasks in ontology learning. Concepts are important entities in a domain, as they can help to express the semantic content of texts and characterize the documents semantically. This process is considered to be difficult and it is usually carried out by human experts. However, this manual identification of concept tends to be slow and subjective and does not scale-up with larger document collections.

Appendix D shows the acceptance or rejection of the top 50 results from TF-IDF and C -Value methods by the linguist. The value beside each concept denotes the likelihood of each one being a valid term, namely its TF-IDF and C-Value measures. The linguist evaluated the top 50 concepts extracted by these two methods; and a precision of 70% is achieved by a multi word concept extractor module and 84%by a single word concept extractor module. Formulating an ontology lexicon with sensible concepts is an important prerequisite for the determination of relations that model the domain appropriately so this module is very important to the whole system.

5.3. TAXONOMIC RELATIONS MINER

This topic discusses the testing for the taxonomic relationships mining module. As discussed in the previous section, the results from testing is measured using precision. But we believe that the advantage of the methods and tools designed should not only be evaluated in terms of their precision or recall, but the time it saves and the computational overhead it has simplified should also be considered.

The agglomerative clustering method that we have employed gives us a plain numerical value denoting the similarity between two clusters and this will help us choose the most important ones among the results obtained. All the relations are labeled as 'is a' (sub-class to super-class) relationships.

We forward the100 concepts (50 single word and another 50 multi-word concepts) that are extracted by the previous module to the taxonomy builder. It was able to build 47 distinct relations, among which 14 were vague and results in a precision of 70.21%. But the module was supposed to get more hierarchical relationships than those that are found, which can be improved by using another method that could extract hierarchies that could be extracted by a technique other than lexical similarity.

5.4. NON-TAXONOMIC RELATIONS MINER

The non-taxonomic relation mining module works using a verbal expression as a core relationship indicator in a sentence. As discussed in Section 3.5.1a verb in a sentence is assumed to potentially represent a relation between the subject and object of that sentence. Whenever we have two concepts in a sentence this module assumes that the two concepts are related in a way that the first concept to appear in a sentence is a domain and the ones that follow are range in that relation (the relation that is described by the verb that comes after both concepts).This module also works on discovering the appropriate generalization level for the relations that appear in the corpus, with respect to a given domain taxonomy. From an ontology point of view, it is very important to discover the appropriate generalization level for a relation.

Our method processes every recognized relationship instances, so that only those that appropriately represent a domain are filtered out. We applied the conditional probability measure in order to find the correct level of generalization in the concept hierarchy. A total of 29 non-taxonomic relations were filtered out and out of which 14 were vague/mistaken, which shows that our module has a precision of 51.72%. The performance of this module is small due to the fact that those relations (non-taxonomic) are not occurring in a coherent pattern.

The performance of all modules, in precision, is summarized in Table 5.1.

Table 5.1: Summary of the precision of all modules

Module	Algorithm	Precision Each	Precision Average
Concept Extraction	TF-IDF	84%	77%
	C-Value	70%	
Taxonomy Building	Hierarchical Agglomerative Clustering	70.21%	70.21%
Non-taxonomic relations mining	Using verbal Expressions	51.72%	51.72%

CHAPTER SIX: CONCLUSION AND FUTURE WORK

6.1. CONCLUSION

This research focused on methods for automatic ontology learning from un-structured Amharic texts and we proposed different methods to be applied at different layers in ontology development process. Extracting all possible concepts that are believed to describe a domain is done first then the concept hierarchy or taxonomy of concepts is constructed which is then enhanced with non-taxonomic relations that are found in a corpus. Concepts of any kind (single word and multi word) and any relationship that could be deduced from the corpus are recognized by our modules at different stages.

We let our system extract both single word and multi word concepts rather than considering only one of those kinds. This will help the ontology engineer to model a domain in a more solid and comprehensive way. TF-IDF and C-Value methods are used to extract single word and multi word term from a corpus respectively. TD-IDF is a statistical method whereas C-Value is a method that combines linguistic and statistical approaches.

The backbone of the domain Ontology, the taxonomic tree, is constructed using the hierarchical agglomerative clustering technique. The taxonomy is then transformed into a full Ontology when it is enriched with non-taxonomic relations. In this research, non-taxonomic relations are extracted by considering a verbal phrase in a sentence as a main drawer of relation between the two noun phrases. The concept that comes first is considered as domain and concepts that follow are considered as a range to that relation.

Once all concepts and the possible relationships that exist among them are extracted the domain is represented by ontology in OWL that is created on protégé, as discussed in Section 3.6.1 and can be manipulated later using jena semantic web framework using the generated code, see Appendix E.

The modules in Amharic Ontology Learner were developed mostly from scratch. Though there were previous researches on Amharic NLP, a Full-fledged and working Amharic NLP tools were not available which rammed us to develop our own tool. E.g. had there been a working Amharic POS tagger we would have been able to test our modules with highly domain specific corpus, but we couldn't find one which makes us test our tools on a tagged news corpus.

The evaluation is made on the three main components of Amharic ontology learner, concept extractor, taxonomy generator and non-taxonomic relationships miner and a precision of 77%, 20.21% and 51.72 % is achieved respectively.

6.2. FUTURE WORK

During the course of this research, we were able to find some issues that we believe merit further investigation. These future explorations are important to make the ontology learning process more efficient and full-fledged. In this regard we recommend the following to be considered by future researchers:

- The methods explored in this research can extract a concept that is explicitly represented by a word or words in a document, but concepts may exist in a document without being explicitly symbolized by a word or words. E.g. a paragraph discussing about Tourism may not have the word Tourism in it, but the concept 'Tourism' needs to be discovered. So a semantic analysis on a text is needed to deduce such a concept from a description.
- The taxonomy construction can be improved by adding a statistical module on top of the already designed taxonomy builder. This statistical module shall have a manually constructed taxonomy seeds, from which it learns to find other concepts in a document with similar pattern. The initial seed can be grown through time with the taxonomies found from the document and approved by the knowledge engineer to be correct.

This will make the taxonomy better in terms of correctness and completeness.

- Our work is limited to recognizing concepts and the relations among them (Taxonomic and non-taxonomic), but the ontology could be much improved if different attributes and instances of those concepts and relations are included. This can be done by manually constructing an Amharic lexical dictionary like Wordnet in English. These attributes are very important for driving important Axioms and it will also help when using the ontology in other applications. E.g. (1) A concept Tiger (ነብር) can have attributes; has_claws (ጥፍርአለው) and can_run_fast (በፍጥነት ይሮጣል) (2) Concepts wild_animal(የዱር እንሰሳ) and domestic_animal(የቤት እንሰሳ) can have a constraint that keeps them as mutually exclusive, a concept in one of those classes cannot belong to the other.
- Furthermore our Amharic ontology learner can be extended to support cardinality constraints. E.g. Defining relations as symmetric or transitive and equivalence for concepts. The theoretically discussed advantages of ontology can also be shown by developing a system that uses an ontology input and prove the advantages.

REFERECES

[1] Ochoa, José, Maria Luisa Hernandez-Alcaraz, Rafael Valencia and Rodrigo Martinez, A Semantic Role-Based Approach for Ontology Learning from Spanish Texts, *International Symposium on Distributed Computing and Artificial Intelligence,* Springer Berlin/Heidelberg, 2011.

[2] Biemann, Chris, Ontology learning from text: A survey of methods, *LDV forum*, Vol. 20. No. 2, 2005.

[3] Drumond, Lucas, and Rosario Girardi, A survey of ontology learning procedures,*3rd Workshop on Ontologies and Their Applications (WONTO 2008)*, Salvador, Brasil,2008.

[4] Maedche, Alexander, and Steffen Staab, Ontology learning for the semantic web, *Intelligent Systems*, IEEE 16.2 (2001): 72-79.

[5]Mindaye, Tessema, and Solomon Atnafu, Design and Implementation of Amharic Search Engine, *Signal-Image Technology & Internet-Based Systems (SITIS), 2009 Fifth International Conference on*, IEEE, 2009.

[6] García-Sánchez, Francisco, Jesualdo Tomas, Rafael Valecia, Juan Miguel and Rodrigo Martinez, Combining Semantic Web technologies with Multi-Agent Systems for integrated access to biological resources, *Journal of biomedical informatics* 41.5 (2008): 848.

[7] Happel, Hans-Jörg, and Stefan Seedorf, Applications of ontologies in software engineering, *Proc. of Workshop on Sematic Web Enabled Software Engineering*(SWESE) on the ISWC, 2006.

[8] Hashim, Fatimah, Gazi Mahabubul Alam, and Saedah Siraj, Information and communication technology for participatory based decision-making-E-management for administrative efficiency in Higher Education,*Int. J. Phys. Sci* 5.4 (2010): 383-392.

[9] Maedche, Alexander, and Steffen Staab, The text-to-onto ontology learning environment, *Software Demonstration at ICCS-2000-Eight International Conference on Conceptual Structures*, 2000.

[10] http://www.omniglot.com/writing/amharic.htm, Feb 22, 2012

[11] Zhang, Ziqi, Jose Iria, Christopher Brewter and Fabio Ciravegna, A comparative evaluation of term recognition algorithms, *unpublished,*(2008).

[12]Bourigault, Didier, and Christian Jacquemin, Term extraction+ term clustering: An integrated platform for computer-aided terminology, *Proceedings of the ninth conference on*

European chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 1999.

[13] Kruk, Sebastian Ryszard, Berhnhard haslhofer, Piotr Piotrowski, Adam Westerskiand Tomasz Woroniecki, The role of ontologies in semantic digital libraries, *NkOS Workshop*, Vol.1, 2007.

[14] DiLeo, Jonathan, Timothy Jacobs, and Scott DeLoach, Integrating ontologies into multiagent systems engineering, Air Univ Maxwell Afb Al Center For Aerospace Doctrine Research And Education, 2006.

[15] Brewster, Christopher, and Kieron O'Hara, Knowledge representation with ontologies: the present and future, *Intelligent Systems*, IEEE 19.1 (2004): 72-81.

[16] Wang, Xiao Hang, Tao Gu, Da Qing Zhang and Hung Keng Pung, Ontology based context modeling and reasoning using OWL, *Pervasive Computing and Communications Workshops, 2004, Proceedings of the Second IEEE Annual Conference on*, IEEE, 2004.

[17] Buitelaar, Paul, Philipp Cimiano, and Bernardo Magnini, Ontology learning from text: methods, evaluation and applications, Vol. 123, Ios PressInc, 2005.

[18] Buitelaar, Paul, Daniel Olejnik, and Michael Sintek, A protégé plug-in for ontology extraction from text based on linguistic analysis, *The Semantic Web: Research and Applications* (2004): 31-44.

[19] Getahun Amare, *ዘመናዊ የአማርኛ ሰዋሰው በቀላል አቀራረብ*, *Alpha printers*, Addis Ababa, 1989ዓ .ም,

[20] Baye Yimam, *አጭርና ቀላል የአማርኛ ሰዋሰው*, *Alpha Printers*, Addis Ababa, 2002ዓ .ም

[21] Frantzi, Katerina, Sophia Ananiadou, and Hideki Mima, Automatic recognition of multi-word terms: the C-value/NC-value method, *International Journal on Digital Libraries* 3.2 (2000): 115-130.

[22] Nakagawa, Hiroshi, Experimental evaluation of ranking and selection methods in term extraction, Bourigault D, L'Homme M.-C, Jacquemin C.(éd.), *Recent advances in computational terminology, John Benjamins Publishing Company*, Amsterdam (2001): 303-326.

[23] Ananiadou, Sophia, A methodology for automatic term recognition, *Proceedings of the 15th conference on Computational linguistics-Volume 2,* Association for Computational Linguistics, 1994.

[24] Bourigault, Didier, Surface grammatical analysis for the extraction of terminological noun phrases, *Proceedings of COLING*, Vol. 92, 1992.

[25] Ramos, Juan, Using tf-idf to determine word relevance in document queries, *Proceedings of the First Instructional Conference on Machine Learning*, 2003.

[26]Antoniou, Grigoris, and Frank van Harmelen, Web ontology language: Owl, *Handbook on ontologies* (2009): 91-110.

[27] Cimiano, Philipp, Johanna Völker, and Rudi Studer, Ontologies on Demand? A Description of the State-of-the-Art, Applications, Challenges and Trends for Ontology Learning from Text, 2006.

[28] Dunning, Ted, Accurate methods for the statistics of surprise and coincidence, *Computational linguistics* 19.1 (1993): 61-74.

[29] Nakagawa, Hiroshi, Automatic term recognition based on statistics of compound nouns, *Terminology* 6.2 (2001): 195-210.

[30] Cimiano, P., M. Hartung, and E. Ratsch, Finding the appropriate generalization level for binary ontological relations extracted from the Genia corpus, *Proceedings of the International Conference on Language Resources and Evaluation (LREC),* 2006.

[31] Hindle, Donald, Noun classification from predicate-argument structures, *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 1990.

[32] Cimiano, Philipp, and Johanna Völker, Text2Onto,*Natural Language Processing and Information Systems* (2005): 257-271.

[33] Faure, David, and Claire Nédellec, A corpus-based conceptual clustering method for verb frames and ontology acquisition, *LREC workshop on adapting lexical and corpus resources to sublanguages and applications*, 1998.

[34] Caraballo, Sharon A, Automatic construction of a hypernym-labeled noun hierarchy from text, *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, Association for Computational Linguistics, 1999.

[35] Poesio, Massimo, Tomonori Ishikawa, Sabine Schulte and Renata Viera, Acquiring lexical knowledge for anaphora resolution, *Proceedings of the 3rd Conference on Language Resources and Evaluation*, Vol. 4, 2002.

[36] Berland, Matthew, and Eugene Charniak, Finding parts in very large corpora, *Annual Meeting-Association For Computational Linguistics, Vol. 37, ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, 1999.

[37] Bisson, Gilles, Claire Nédellec, and L. Canamero, Designing clustering methods for ontology building-The Mo'K workbench, *Proceedings of the ECAI Ontology Learning Workshop,* 2000.

[38] Hearst, Marti A, Automatic acquisition of hyponyms from large text corpora, *Proceedings of the 14th conference on Computational linguistics-Volume 2, Association for Computational Linguistics*, 1992.

[39] Daille, Béatrice, Study and implementation of combined techniques for automatic extraction of terminology, *the balancing act: Combining symbolic and statistical approaches to language* 1 (1996): 49-66.

[40] Wermter, Joachim, and Udo Hahn, Finding new terminology in very large corpora, *Proceedings of the 3rd international conference on Knowledge capture*, ACM, 2005.

[41] Maynard, Diana, and Sophia Ananiadou, Trucks: a model for automatic multiword term recognition, *Journal of Natural Language Processing* 8.1 (2000): 101-126.

[42] Vivaldi, Jordi, 2Lluís Màrquez, and Horacio Rodríguez, Improving term extraction by system combination using boosting, *Machine Learning: ECML* 2001 (2001): 515-526.

[43] Agrawal, Rakesh, Tomasz Imieliński, and Arun Swami, Mining association rules between sets of items in large databases, *ACM SIGMOD Record*, Vol. 22, No. 2, ACM, 1993.

[44] Dagan, Ido, and Ken Church, Termight: Identifying and translating technical terminology, *Proceedings of the fourth conference on Applied natural language processing, Association for Computational Linguistics*, 1994.

[45] Nenadić, Goran, Irena Spasić, and Sophia Ananiadou, Automatic discovery of term similarities using pattern mining, *COLING-02 on COMPUTERM 2002: second international workshop on computational terminology-Volume* 14, Association for Computational Linguistics, 2002.

[46] Berners-Lee, Tim, James Hendler, and Ora Lassila, The semantic web, Scientific american 284.5 (2001): 28-37.

[47] Gil-García, Reynaldo, Jose M. Badia-Contelles, and Aurora Pons-Porrata, A general framework for agglomerative hierarchical clustering algorithms, Pattern Recognition, 2006, ICPR 2006. 18th International Conference on, Vol. 2, IEEE, 2006.

[48] Powers, David MW, Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation, School of Informatics and Engineering, Flinders University, Adelaide, Australia, Tech. Rep, SIE-07-001 (2007).

[49] Berners-Lee, Tim; Fischetti, Mark, Weaving the Web, Harper San Francisco, chapter 12. ISBN 9780062515872, 1999

[50] Vossen, Piek, EuroWordNet: a multilingual database for information retrieval, Proceedings of the DELOS workshop on Cross-language Information Retrieval, 1997.

[51] Miller, George A, WordNet: a lexical database for English, Communications of the ACM 38.11 (1995): 39-41.

Appendix A--(Python Code: Sentence splitter)

```
import codecs
import re
import math
import string
AllLines = []
me = []
cur = []

def readfile():                                  ## Read file Function Definition
   f = open(r'c:\My thesis\Corpus\My original.txt','r',encoding="utf8")
for line in f:
temp = str.split(line)
        i = len(temp)
        k = 0
for j in range(0, i):                    ## Check for every Newline
if temp[j] == '።':
fin = ' '.join(temp[k:j+1])
AllLines.append(fin)
                k = j+1

if j == i:
fin = ' '.join(temp[k:j])
AllLines.append(fin)
f.close()
print ('Successfully finished reading the file')

def prinall():                                   ## Write every line separately on another file
for further process
   k = open(r'c:\My thesis\\Corpus\Annotated.txt','w',encoding="utf8")
for item in AllLines:
k.writelines(item + '\n')
k.close()
print ('Annotation was Successfull ')
readfile()

prinall()
```

Appendix B --(Python Code: POS recognizer for linguistic filtering)

```
import codecs
import re
import math
AllLines = []
All = []
me = []
candidate = []
current = []

def readfile():                    ### Read Input FIle
    f = open(r'D:\Files\My
thesis\Corpus\Clustered\OTourism.txt','r',encoding="utf8")
for line in f:
AllLines.append(line)
f.close()

def Linguisticfilter(this,that):           ### Recognize anything between , this(<)
and that (>)
       Ntitle = 0
for me in AllLines:
           i = len(me)
           current = ""
for j in range(0, i-1):
if me[j] == this:
                  #print('J')
                  #print(j)
for k in range(j+1,i):
if me[k] == that:
                        item =""
                        #print('K')
                        #print(k)
for temp in range(j+1,k):
item = item  +me[temp]
                        #print(item)
firstp = 0
if item == 'title':
                           j = k+1
```

```
current = current + "<" + item + "> "
                            Ntitle +=1
                        if  item == 'PREP' or item == 'PUNC' or item == 'PRONP' or
item == 'AUX' or item == 'PRON' or item == 'V' or item == 'VP' or item == 'VC' or
item == 'VREL' or item == 'VN':
break
else:
                            manew = ""
for checker in range(j,0,-1):
if me[checker] == that:
firstp = checker +1
break
for temp in range(firstp,j-1):
manew = manew  +me[temp]
                            #print(manew)
current = current + manew
break
candidate.append(current)
print (Ntitle)

def prinall():                              ### Print the result after filtering
    k = open(r'D:\Files\My
thesis\Corpus\Clustered\CTourism.txt','w',encoding="utf8")
for item in candidate:
k.writelines(item + '\n')
k.close()
print ('Text is ready')

readfile()
Linguisticfilter('<','>')
prinall()
```

Appendix C-- (Stop words lists)

{ "እዚህ", "እዚያ","ከ", "ናቸው", "ትናንት","ጥቂት", "በርካታ", "ብቻ","ሁሉም", "ሌላ", "ሌሎች","ሁሉ",
"እያንዳንዱ","እያንዳንዳችው","እያንዳንዷ","እንደገና", "ማንም","እባክዎ", "እባክሽ","እባክህ", "ተጨማሪ",
"ውጪ", "ናት","ነበሩ", "ነበረች", "ያ","ነገሮች", "ከፊት","ከላይ", "ታች", "ከታች","በታች", "የታች","ከውስጥ",
"በውስጥ","የውስጥ", "ጎላ", "ከጎላ","የጎላ", "መካከል","ከመካከል", "ሰሞኑን","ከሰሞኑ",
"በሰሞኑ","የሰሞኑ","ጋራ","የጋራ", "ከጋራ","ተለያዩ", "ተለያየ","ድረስ", "እስከ", "በጣም","ግን", "ሲሆን",
"ሲል","ውስጥ", "ላይ", "ነይ","ነው", "ጋር", "ናቸው","ይህ", "ወደ","ወዘተ", "እና", "ወይም","እንደ", "ፊት",
"ወደፊት", "ነገር","በጎላ", "በኩል","ስለ", "ደግሞ", "እንጂ","እንዲሁም" }

Appendix D --(Concepts Inspection by the linguist)

Single word concepts

Rank	TF-IDF	Concept	Remark
1	0.431565	ዱር	✓
2	0.42282	ፓርክ	✓
3	0.415888	ፖርክ	✓
4	0.347803	ቅርስ	✓
5	0.339642	ስሳት	✓
6	0.311916	ቆርክ	✓
7	0.286702	ሃላፍ	✓
8	0.277259	ዝርይ	✓
9	0.277259	ሙዚይ	✓
10	0.270327	አእዋፍ	✓
11	0.263396	ተራር	✓
12	0.207944	አጥብ	X
13	0.207944	ስኮትላንድ	✓
14	0.207944	ሱብ	✓
15	0.194081	ኑበር	✓
16	0.192701	ደን	✓
17	0.18715	አቡን	X
18	0.18715	ታቦት	✓
19	0.18715	መናገሸ	✓
20	0.180218	መቅደል	✓
21	0.180218	ላሊበል	✓
22	0.176549	ጎብኝ	✓
23	0.173901	ባለሙይ	✓
24	0.156933	እድም	X
25	0.156933	ቀበር	✓
26	0.155101	አካባብ	✓
27	0.155101	ተፈጥር	✓
28	0.145561	አእዋፋት	✓
29	0.140964	ቢር	✓
30	0.132334	ቱሪስት	✓
31	0.124766	ድኩል	✓
32	0.124766	ደንቆር	✓
33	0.124766	ተፈጥሮአው	✓
34	0.124766	ስው	✓
35	0.124766	ርል	X
36	0.120178	ቱሪዝ	✓
37	0.117835	ትበቅ	X
38	0.117501	መምሪይ	✓
39	0.110904	ጉዳም	✓

40	0.109496	ክልል	✓
41	0.106442	ገለጻት	X
42	0.103972	ዝንጀር	✓
43	0.103972	ዝህ	✓
44	0.103972	ቤተክርስቲይ	✓
45	0.103972	ቁም	X
46	0.103972	ሰፋር	✓
47	0.103972	ሙዚየም	✓
48	0.103972	ሆም	X
49	0.097812	መስህብ	✓
50	0.097041	ዩኔስክ	✓

Multi-word Concepts

Rank	Concept	C-Value	Remark
1	እንፎርሜሽን ማእከል	19.5	✓
2	የዱር እንስሳት	17.33333	✓
3	ለዋልታ እንፎርሜሽን ማእከል	11.69925	✓
4	ዋልታ እንፎርሜሽን ማእከል	10.61429	✓
5	ባህል ቱሪዝምና ማስታወቂያ	4.509775	✓
6	ተራሮች ብሄራዊ ፓርክ	4.094738	✓
7	ባህል ማስታወቂያና ቱሪዝም ቢሮ	4	✓
8	ባህል ቱሪዝምና ማስታወቂያ ቢሮ	4	✓
9	የዞኑ ንግድ እንዱስትሪና ቱሪዝም መምሪያ	3.965784	✓
10	ተራሮች ብሄራዊ	3.5	X
11	ባህል ማስታወቂያና ቱሪዝም	3.424813	✓
12	ንግድ እንዱስትሪና ቱሪዝም	3.08985	✓
13	የዞኑ ንግድ እንዱስትሪና ቱሪዝም	3	✓
14	ንግድ እንዱስትሪና ቱሪዝም መምሪያ	3	✓
15	የዱር እንስሳት ከፍተኛ ባለሙያ	3	✓
16	ብሄራዊ ፓርክ	3	✓
17	በቢሮው የባህል መምሪያ ሃላፊ	3	✓
18	የቅርስ ትናትና ትቢቃ ባለስልጣን	3	✓
19	የኢትዮጵያ ቱሪዝም ኮሚሽን	2.924813	✓
20	ለዋልታ ኢንፎርሜሽን ማእከል	2.924813	✓
21	የቅርስ ትናትና ትቢቃ	2.83985	X
22	ቱሪዝም ኮሚሽን	2.666667	✓
23	መምሪያ ሃላፊ	2.666667	✓
24	ኢንፎርሜሽን ማእከል	2.666667	✓
25	የአለም ቱሪዝም ቀን	2.33985	X
26	የጽህፈት ቤቱ ሃላፊ	2.33985	✓
27	ውቅር አብያተ ክርስቲያናት	2.33985	✓
28	ማስታወቂያና ቱሪዝም ቢሮ	2.33985	✓
29	ቱሪዝምና ማስታወቂያ ቢሮ	2.33985	✓
30	የቀይ ቀበሮዎች ቁጥር	2.33985	X

31	የዞኑ ንግድ እንዱስትሪና	1.754888	X
32	እንዱስትሪና ቱሪዝም መምሪያ	1.754888	✓
33	ቅዬ የዱር እንስሳት	1.754888	✓
34	የዱር እንስሳት ከፍተኛ	1.754888	X
35	እንስሳት ከፍተኛ ባለሙያ	1.754888	X
36	የሱፍ አብዱላሂ ሱከር	1.754888	✓
37	የዱር እንስሳት እና	1.754888	X
38	ቅርሶችን የመኑበኘት ባህሉ	1.754888	X
39	የአዲስ አበባ ሙዚየም	1.754888	✓
40	የሰሜን ተራሮች ብሄራዊ	1.754888	✓
41	በቢሮው የባህል መምሪያ	1.754888	✓
42	የባህል መምሪያ ሃላፊ	1.754888	✓
43	እንፎርሜሽን ማእከል መግለጫ	1.754888	X
44	በመቅደላ ቶርነት ወቅት	1.754888	X
45	ትናትና ትቢቃ ባለስልጣን	1.754888	X
46	የባሌ ተራሮች ብሄራዊ	1.754888	✓
47	የበጹእ ወቅዱስ ፓትሪያርክ	1.754888	X
48	የቱሪዝም አማካሪ ምክር	1.754888	X
49	የድንጋይ ዘመን ሰዎች	1.754888	X
50	በቤንሻንጉል ጉሙዝ ክልል	1.754888	✓

Appendix E--(Portion of the code generated by protégé)

```
<?xml version="1.0"?>
<!DOCTYPE Ontology [
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
<!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<Ontology xmlns="http://www.w3.org/2002/07/owl#"

xml:base="http://www.semanticweb.org/ontologies/2013/1/Tourism_taxonomy.owl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"

ontologyIRI="http://www.semanticweb.org/ontologies/2013/1/Tourism_taxonomy.owl"

versionIRI="http://www.semanticweb.org/ontologies/2013/1/Tourism_taxonomy.owl">
<Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
<Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
<Prefix name="" IRI="http://www.w3.org/2002/07/owl#"/>
<Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
<Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
<Declaration>
<Class IRI="#A&amp;apos;ewaf"/>
</Declaration>
<Declaration>
<Class IRI="#Addis_Ababa_Muziyem"/>
</Declaration>
<Declaration>
<Class IRI="#Atibi"/>
</Declaration>
<Declaration>
<Class IRI="#Bahil_Tourism_Ena_Mastawukiya"/>
</Declaration>
<Declaration>
<Class IRI="#Be_Birow_Ye_Bahil_memriya_Halafi"/>
</Declaration>
<Declaration>
```

<Class IRI="#Bebirow_yebahil_memriya"/>
</Declaration>
<Declaration>
<Class IRI="#Benishangul_Gumiz_Kilil"/>
</Declaration>
<Declaration>
<Class IRI="#Biherawi_Park"/>
</Declaration>
<Declaration>
<Class IRI="#Den"/>
</Declaration>
<Declaration>
<Class IRI="#Ensisat"/>
</Declaration>
<Declaration>
<Class IRI="#Ensisat_kefitegna_balemuya"/>
</Declaration>
<Declaration>
<Class IRI="#Ye_Alem_Turizm_Ken"/>
</Declaration>
<Declaration>
<Class IRI="#Ye_Bahil_memriya_Halafi"/>
</Declaration>
<Declaration>
<Class IRI="#Ye_Tsihifet_Bet_Halafi"/>
</Declaration>
<Declaration>
<Class IRI="#Ye_semen_teraroch_biherawi"/>
</Declaration>
<Declaration>
<Class IRI="#Yebale_teraroch_Biherawi"/>
</Declaration>
<Declaration>
<Class IRI="#Yedur_Ensisat"/>
</Declaration>
<Declaration>
<Class IRI="#Yedur_Ensisat_kefitegna_Balemuya"/>
</Declaration>
<Declaration>
<Class IRI="#Ziriya"/>
</Declaration>
<Declaration>
<ObjectProperty IRI="#Alemekahed"/>
</Declaration>
<Declaration>
<ObjectProperty IRI="#Asreditewal"/>

```
</Declaration>
<Declaration>
<ObjectProperty IRI="#Endemigegnu"/>
</Declaration>
<Declaration>
<ObjectProperty IRI="#Endemizegaj"/>
</Declaration>
<Declaration>
<ObjectProperty IRI="#Endeneberu"/>
</Declaration>
<Declaration>
<ObjectProperty IRI="#Endimedebulet"/>
</Declaration>
<Declaration>
<ObjectProperty IRI="#Meberaket"/>
</Declaration>
<Declaration>
<ObjectProperty IRI="#Megobgnet"/>
</Declaration>
<Declaration>
<ObjectProperty IRI="#Metebek"/>
</Declaration>
<SubClassOf>
<Class IRI="#A&amp;apos;ewaf"/>
<Class IRI="#Tourism"/>
</SubClassOf>
<SubClassOf>
<Class IRI="#Addis_Ababa_Muziyem"/>
<Class IRI="#Muziyem"/>
</SubClassOf>
<SubClassOf>
<Class IRI="#Atibi"/>
<Class IRI="#Tourism"/>
</SubClassOf>
<SubClassOf>
<Class IRI="#Bahil_Tourism_Ena_Mastawukiya"/>
<Class IRI="#Tourism"/>
</SubClassOf>
<SubClassOf>
<Class IRI="#Be_Birow_Ye_Bahil_memriya_Halafi"/>
<Class IRI="#Ye_Bahil_memriya_Halafi"/>
</SubClassOf>
<SubClassOf>
<Class IRI="#Bebirow_yebahil_memriya"/>
<Class IRI="#Tourism"/>
</SubClassOf>
```

<ObjectPropertyDomain>
<ObjectProperty IRI="#Endemizegaj"/>
<Class IRI="#Park"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
<ObjectProperty IRI="#Endeneberu"/>
<Class IRI="#Den"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
<ObjectProperty IRI="#Endimedebulet"/>
<Class IRI="#Suba"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
<ObjectProperty IRI="#Meberaket"/>
<Class IRI="#Korke"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
<ObjectProperty IRI="#Megobgnet"/>
<Class IRI="#Tuizm"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
<ObjectProperty IRI="#Metebek"/>
<Class IRI="#Turizm"/>
</ObjectPropertyDomain>
<ObjectPropertyRange>
<ObjectProperty IRI="#Alemekahed"/>
<Class IRI="#Turizm"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
<ObjectProperty IRI="#Asreditewal"/>
<Class IRI="#Turizm"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
<ObjectProperty IRI="#Endemizegaj"/>
<Class IRI="#Turizm"/>
</ObjectPropertyRange>
</Ontology>

<!-- Generated by the OWL API (version 3.2.5.1912)
http://owlapi.sourceforge.net -->

Declaration

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

Declared by:

Name: **Berhanu Mengiste**

Signature: _____

Date: _____

Confirmed by advisor:

Name: **Dr Fekade Getahun**

Signature: _____

Date: _____

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