ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCES

Amharic Open Information Extraction

Seble Girma Fisseha

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

Addis Ababa, Ethiopia

March 2020

ADDIS ABABA UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE

Seble Girma Fisseha

Advisor: Yaregal Assabie (PhD)

This is to certify that the thesis prepared by *Seble Girma*, titled: *Amharic Open Information Extraction* and submitted in partial fulfillment of the requirement for the Degree of Master of Science in Computer Science complies with the regulation of the university and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	Name	Signature	Date
Advisor:	Yargal Assabie (PHD)		
Examiner:	Solomon Atnafu (PHD)		
Examiner:	Solomon Gizaw (PHD)		

ABSTRACT

Open Information Extraction is the process of discovering domain-independent relations by providing ways to extract unrestricted relational information from natural language text. It has recently received increased attention and applied extensively to various downstream applications, such as text summarization, question answering, and informational retrieval. Although a lot of Open Information Extraction systems have been developed for various natural language text, no research has been conducted yet for the development of Amharic Open Information Extraction (AOIE).

As litrature has shown, the rule-based approach operating on deep parsed sentences yields the most promising results for Open Information Extraction systems. However, to the best of our knowledge, there is no fully implemented deep syntactic parser available for Amharic language. Therefore, in this thesis, we propose the development of a rule-based AOIE system that utilizes shallow parsed sentences.

The proposed system has six components: Preprocessing, Morphological Analysis, Phrasal Chunking, Sentence Simplification, Relation Extraction, and Post-processing. In the Preprocessing, each word in the input text is labeled with an appropriate POS tag, and then well-formed and informative sentences are filtered out for further processing based on POS tags of words. The Morphological Analysis component produces morphological information about each word of input sentences. The phrasal chunking component divides the input sentence into non-overlapping phrases based on POS and morphological tags of words. The Sentence Simplification component segments the sentence into a number of self-contained simple sentences that are easier to process. In the Relation Extraction, relation instances are extracted from those simplified sentences and finally the post-processing components prints extracted relations in N-ary format.

The proposed method and algorithms were implemented in prototype software and evaluated with a dataset from different domains. In the evaluation, we showed that the system achieved an overall precision of 0.88.

Keywords: Open information extraction, chunking, sentence simplification, relation extraction.

ACKNOWLEDGMENTS

First of all, I would like to thank and praise the Almighty God for his profound love and guidance, and for making this thesis possible.

Secondly, I would like to express my deepest gratitude to my advisor, Dr. Yaregal Assabie for his guidance, unbounded support and his endeavors to make this work complete. I am especially grateful for the freedom he gave me during this time. Without his attentive guide, invaluable trust, and constructive criticism, this thesis can not be achieved.

I also would like to thank my father Girma Fisseha and my mother Achamyelesh Zewde for always encouraging me and supporting my decisions.

Finally, I am grateful to my siblings and friends who encourage me to progress in every aspect.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Acronyms and abbreviations

ACE	Automatic Content Extraction
AOIE	Amharic Open Information Extraction
EE	Event Extraction
IE	Information Extraction
KBP	Knowledge Base Population
MUC	Message Understanding Conferences
NER	Named Entity Recognition
NLP	Natural Language Processing
NP	Noun Phrase
OIE	Open Information Extraction
POS	Part of Speech Tagger
RE	Relation Extraction
TAC	Text Analysis Conferences
TAM	Tense Aspect Mood

CHAPTER ONE: INTRODUCTION

1.1 Background

With the increasing number of digital data sources and the rapid growth of data volume, there has been an increase in interest for utilizing extracted information in everyday applications [1]. Government agencies, research institutions, corporations, and even individuals are all realizing the value of information in free text to make well-informed decisions and to maintain a successful business [2].

Due to the diverse nature of data contained within digital data sources, searching and finding appropriate information is becoming extremely challenging and it remains an open research problem [1]. What makes information extraction difficult is the fact that data is initially unstructured and is described using human-understandable language, which makes the data limited in the degree in which it is machine-interpretable [1]. This problem drived the automation of information and knowledge extraction. A lot of researches are still being conducted to automate the extraction of information from the raw text in order to organize the text in a structured format [3].

Information Extraction (IE) is the task of automatically extracting knowledge from text [4]. The primary goal of designing IE systems is to organize unstructured and semi-structured representation of data into structured representations, such as templates and database entries which makes further analysis or processing of this information easier because computers are much better at processing structured information than they are at processing unstructured information [2].

The core task of IE systems is to identify entities and relationships expressed using natural language. However, the traditional paradigms of IE, which were initially proposed in the series of Message Understanding Conferences (MUC) [4], require either hand-tagged training examples for each target relation or pre-specified relations along with hand-crafted extraction rules as input. Such inputs are specific to the target domain; shifting to a new domain requires extensive human involvement in creating new extraction patterns or specifying hand-tagged new training examples [5]. This approach to IE is not portable across domains and does not

scale to massive and heterogeneous corpora like Web where the relations are unanticipated [6]. To overcome these limitations, Open Information Extraction (OIE) has become more strongly suggested. It has made possible to process massive text corpora without restriction to extract a certain type of relations and attributes, and without having to require much human effort [7].

The OIE paradigm was introduced by Banko *et al.* [5] aiming to develop domain-independent extractors of information by providing ways to extract unrestricted relational information from text. According to [8, 9], OIE has several advantages over the traditional IE approaches. It made easier to extract many kinds of relations without requiring manual labor for building extraction rules and hand-tagged training examples for each relation. Because of its ability to extract information for all relations at once without having them named explicitly, it also has a significant scalability advantage over previous IE architectures. Traditional IE systems usually search for entities that are associated with the type of relation which the system was configured to extract; whereas an OIE system tries to find relations as well as the entities taking part in those relations which are not predefined. Traditional IE systems require a specific pattern for each relation. On the other hand, OIE systems need a set of patterns that are not related to any specific relation, and these features are useful to extract relations of any nature.

Moreover, according to the classification of IE given by Romadhony *et al.* [10], there are two approaches to IE: template-based and relation-discovery approach. Most of the traditional IE systems were using template-based approaches. These approaches require predefined template slots to perform IE tasks. On the contrary, OIE is a relation-discovery approach. Unlike template-based, relation-discovery approaches do not require predefined template slots; rather, they are designed to extract unrestricted types of relations and they represent the discovered relations in the form of set of triples, {(*arg1, rel, arg2*)}, where *arg1* and *arg2* are entities and *rel* is a textual fragment denoting a semantic relation between the two entities. In spite of their limitation of covering broader domain, templates-based approaches can extract a richer and more readable representation of a particular domain than relation-discovery because they extract only relevant information of interest from given text [11]. For that reason, some researchers (e.g., Romadhony *et al.* [10] and Balasubramanian *et al.* [12]) have

investigated methods of using OIE results to extract IE templates automatically. These works showed that template structure can be created automatically from outputs of OIE systems.

The OIE paradigm has achieved a notable measure of success on massive and open-domain corpora drawn from the Web [13]. Nowadays, OIE is extensively being applied to various applications as an important intermediate step of the Natural Language Processing (NLP) stack for many text mining tasks [14]. It has been used for text summarization [15, 16, 17, 18], ontology population [19], event schema induction (template extraction) [10, 12], question answering [20, 21, 22], etc. In addition, according to [6], the output of OIE systems has been used to support tasks like learning selection preferences [23], acquiring common-sense knowledge [24], and recognizing entailment rules [25, 26].

Amharic language, in a variety of respects, has different grammatical structures to other languages like English. Although some efforts are made toward designing IE models for Amharic text, all of these works have used traditional approaches to IE. To the best of our knowledge, no research works have been done on Amharic OIE yet. Therefore, in this thesis, we propose to design an Amharic OIE system specially designed to meet the characteristics of Amharic language.

1.2 Motivation

Amharic is an Afro-Asiatic language of the Semitic branch. It is the second most spoken Semitic language in the world after Arabic [27]. The language serves as the official working language of Ethiopia and it is also the official or working language of some of the states within the federal system [28].

There is much Amharic text available online, which discuss all possible topics, written and published every day by news agencies, governmental organizations, private companies, bloggers, and users of Social Medias. With the dramatic increasing of these digital data collections of different text types such as news messages, articles and web pages, automatic support for the extraction of relevant information from these numerous unstructured texts has become a critical need. In spite of the relatively large number of speakers, Amharic is still a language for which very few computational linguistic resources have been developed, and very little has been done in terms of making useful computer-based applications available to

those who speak Amharic [28]. These gaps produce the drive and desire which motivates us to engage in this research work.

1.3 Statement of the Problem

Traditional IE was relied on a significant amount of human involvement in preparing hand-crafted extraction patterns and hand-tagged training data. In recent years, on the other hand, OIE has become more strongly recommended to overcome the limitations of traditional IE approaches [7]. TextRunner [5], WOE [29], ReVerb [13] and OLLE [30] are some of the most representative examples of OIE systems that offer excellent performance in automatically extracting structured information from unstructured natural language text. These systems have been designed, implemented and evaluated predominantly for English. Unfortunately, because of their heavy reliance on linguistic tools such as part-of-speech taggers and dependency information as well as immediate lexical information to define patterns or constraints for relations, all these systems cannot guarantee the same level of performance on languages other than English. Due to the complexity of natural languages and differences in linguistic characteristics and rules, it is difficult to design a solution that would be universal and applicable to any language. This problem has enticed many researchers to come up with a solution that fits a specific language. As a result, OIE systems for languages other than English are currently being actively researched, e.g., Chinese Open Relation Extraction (RE) [31] and Korean OIE [7].

Because of the advancement in informational technology, the number of Amharic documents in electronic formats on the World Wide Web is rising very rapidly. In order to make use of this growing resource, there is a need to extract useful and structured information automatically. Since OIE systems use language-specific features, systems designed for other languages are not applicable to Amharic. Getasew Tsedalu [32] and Bekele Worku [33] designed IE models for Amharic text. However, both works have focused on small and domain-specific corpora like Newswire articles on infrastructure domain, where the texts are similar and a set of relations are pre-identified. The adaptation of these works to a new domain requires manual labor preparing hand-crafted training examples and extraction patterns. Thus, they are not scalable or portable across domains and not suitable for massive and heterogeneous corpora like Web. Furthermore, both authors used template-based approaches.

In other words, templates were hand-crafted and focused on specific domains of interest which also limits the applicability of IE to a new domain.

To the best of our knowledge, no research works have been done on Amharic OIE yet. The aim of this research work is, therefore, to design an OIE for Amharic text that extracts un-predefined and domain-independent information, and scalable to Web text.

1.4 Objectives

General objective

The general objective of the research work is to design a model for Amharic OIE which can extract open-domain relations and their arguments automatically from Amharic text.

Specific objectives

In order to achieve the general objective stated above, the following specific objectives should be addressed.

- Perform an extensive review on previous research works focusing on IE in general and OIE in particular.
- Collect Amharic text corpus.
- Study the grammatical structure of Amharic language with respect to OIE.
- Design an OIE model for Amharic text corpus.
- Develop a prototype.
- Evaluate the performance of the system.

1.5 Methods

In this research work, we will design an OIE system for Amharic text. Towards achieving the main objective, the following step by step procedures will be followed.

Literature Review

A literature review will be conducted to understand what is done and what is not done in previous works. In this study, to understand the problem domain, works of literature that are directly related to IE will be reviewed and we will give special emphasis to studies conducted on OIE.

Corpus collection and Data preparation

Amharic text data will be collected from different data sources and different facts about Amharic sentence structure will be studied to understand the nature of the Amharic languages with respect to OIE.

Design

Methods used in OIE can be divided into data-based and rule-based. Data-based OIE generates patterns based on training data represented using a dependency tree or PoS-tagged text. Rule-based OIE relies on hand-crafted patterns from PoS-tagged text or rules operating on dependency parse trees. By comparison, rule-based OIE systems achieved much better results. So that the rule-based method will be used in designing OIE for Amharic text.

Prototype Development

In order to evaluate the performance of the proposed method, a prototype system will be developed. Different appropriate tools will be selected and used to develop the prototype. We will try to reuse different existing methods, frameworks and software components as far as possible.

Evaluation techniques

The study includes designing OIE system for Amharic text and implementing a prototype of the system. So, the performance of the system has to be evaluated. At the final stage of the study, the test corpus will be prepared, queries will be constructed and relevance judgment will be made for evaluating the effectiveness of the work.

1.6 Scope and Limitations

The focus of the study is designing OIE system for Amharic language. The scope of the study covers the extraction of information from only text data. Other data types such as video, audio and graphics will not be the focus of the study.

1.7 Application of Results

OIE provides a way to find structured data from open text. The output of OIE can be used in various applications. Some of the applications are the listed below:

- **Question Answering** (QA): aims to automatically answer a user's factual question in natural language by extracting answers from web pages scattered across the internet. QA systems require understanding and structuring the information in a form that can be later mapped to questions and answers. OIE can be used to provide structured information about the content of web pages of any domain.
- **Opinion Mining System:** extracts opinion information about a particular object which is available in blog posts, social media, and different websites. OIE systems can provide structured information about the content of web pages.
- **Ontology Population**: ontology is a collection of facts (i.e. entities and relations between entities). Ontology can be populated either automatically or manually from unstructured or semi-structured data sources. Since OIE systems are able to extract relation instances from a large collection of documents, the output of these systems can be used to populate the ontology of any domain. The applicability of OIE to the ontology population is studied in [19].
- **Text summarization**: relation instances extracted from OIE systems could be used to infer or measure the redundancy between sentences based on the relation instances extracted from the input corpora. The authors in [15, 16, 17] studied ways of generating text summary based on relations that are more relevant to each query.

Therefore, NLP practitioners, researchers, government offices, traders, companies, managers and even individuals will be the users that directly or indirectly benefit from the research work.

1.8 Thesis Outline

The remaining chapters of this thesis are organized as follows. In Chapter 2, we review the basic structure of Amharic language, the different approaches, methods, and challenges in early work on IE and it also provides an overview of OIE and the state-of-the-art approaches used in OIE systems. Chapter 3 discusses related work conducted in the field of OIE. In Chapter 4, we introduce AOIE, the first Amharic OIE system that we proposed to perform open information extraction from Amharic sentences. Chapter 5 describes the experiment and discusses the results. Finally, In Chapter 6, we summarize the achieved results and the conclusions about the performance of the proposed Amharic OIE system.

CHAPTER TWO: LITERATURE REVIEW

In this chapter, literature in Information Extraction (IE) as well as in Open Information Extraction (OIE) are reviewed. First, the Amharic alphabet, word classes, phrases, clauses, and structure of Amharic sentences are discussed. Then, the chapter provides a brief introduction to IE, emphasizing its subtasks and the main classification of IE. The history of IE development and the comparison between traditional IE and OIE can be traced through the discussion. This is followed by a thorough discussion of OIE with the aim of identifying state-of-the-art methods of IE. Finally, approaches and evaluation techniques used in OIE are investigated.

2.1 Amharic Language

Amharic is an official working language of Ethiopia and it is a lingua franca by non-Amharic native speakers in Ethiopian towns. Among the most widely spoken Semitic languages such as Arabic, Tigrinya, Hebrew, Aramaic, Assyrian and Maltese, Amharic is the second most spoken Semitic language in the world next to Arabic [27]. It is also the most widely spoken Sematic language in the Horn of Africa. The majority of the speakers of Amharic are found in Ethiopia, but there are also Millions of emigrants outside of Ethiopia speak the language. Particularly, Most of the Ethiopian Jewish communities in Israel and emigrates in the USA, Canada, and European countries speak Amharic [28].

2.1.1 Amharic Writing System

Amharic is written using Ethiopic script, a version of the Ge'ez script known as ፊደል -Fidel ("alphabet", "letter", or "character") [34]. It is written in a tabular format of seven columns where the first column represents the base characters and others represent their derived vocal sounds. In addition, there are about two scores of characters representing labialized characters such as ሏ (lwa), ሟ (mua), ሷ (sua), etc. There are 33 base characters in Amharic alphabet. Each base character has six other forms which are derived by applying different vowels to the base character. While the base characters have the vowel ኧ(ä), its derived characters have vowels in the order of ኡ(u), ኢ(i), ኣ(a), ኤ (e), እ (), and ኦ (o). For example, for the base character ሰ (sä) the following six characters are derived from the base character: ሱ (su), ሲ (si), ሳ (sa), ሴ (se), ስ (s), ሶ (so). Except from the two base forms (i.e., ኧ and ዐ) which represent vowels (ä),

the rest of 32 base characters are a pair of consonants and vowels. The vowels are not encoded explicitly but appear as modifiers of the base characters. Therefore, these 33 basic characters with their six derived forms will give (33 + 6*33) syllable patterns (syllographs) [33] (See Appendix A).

The Amharic writing system also consists of punctuation marks. Some Amharic punctuation marks are exactly the same as English punctuation marks, but there are also some punctuation marks that are different from their English equivalent [34]. Table 2.1 lists frequently used Amharic punctuation marks, their equivalent in English, and their usage with examples.

Table 2.1: Amharic punctuation marks and their usage

No	Name	Amharic Symbol	English Symbol	Usage	Examples
1	Full stop (Period) አራት ነጥብ	።	.	Marks the end of declarative sentences.	✓ ማርታ መጽሃፍ ገዛች። /*Marta bought a book.*
2	Comma ነጠላ ሰረዝ	፣	,	Show a separation of ideas or elements within the structure of a sentence.	✓ ማርታ መጽሃፍ ፣ ደብተር ፣ ቦርሳ እና ጫማ ገዛች። /*Marta bought book, exercise book, bag and shoes.*
3	Question Marks ጥያቄ ምልክት	?	?	Indicate a direct question when placed at the end of a sentence.	✓ ዩሃንስ መቼ ይመለሳል? /*When will Yuhanis return back?*
4	Exclamation point ቃል አጋኖ	!	!	Used when a person wants to express an emotion or add emphasis.	✓ በጣም ይገርማል! /*Very Amazing!*
5	Semi colon ድርብ ሰርዝ	፤	;	Connect independent clauses.	
6	Quotations Mark ትእምርተ ጥቅስ	“ ”	“ ”	Used to mark quotes	✓ ዩሃንስ “ማርታ መጽሃፍ ገዛች።” አለ ።

2.1.2 Word Classes

The most commonly used Amharic word classes are: Noun, Verb, Adjectives, Adverb, Preposition and Conjuctions [36].

Noun

Nouns are words that are used to name or identify any of the categories of things, people, animals, places, ideas, or a particular of one of these entities [33].

An Amharic noun consists of a stem and zero or more affixes. Despite some exceptional plural forms exist in Amharic (e.g., ‘መንግሥታት mengi^st-at (governments)’ is a plural form of a noun ‘መንግስት mengist （government)’), the most common plural form of Amharic noun suffix is “-ኦች/-ዎች (-och/-woch)”. To convert nouns to their plural form, if a noun ends with consonant, ‘-ኦች (-och)’ will be added to the end of the noun and the vowels ‘O’ changes the structure of the last letter (e.g., ሰዎች (ሰው_ኦች) （sew-och). If a noun ends with a vowel, ‘-ዎች/-woch’ will be added to the end of the noun [34].

There are many possible possessive suffixes for the different combinations of person, number, and gender. Table 2.1 shows how Amharic nouns can be affected for person, number and gender for the noun ‘ወንድም -wendim (Brother)’.

Table 2.1: Amharic noun affection

Person	**Gender**			
	Singular		**Plural**	
	Masculine	**Feminine**	**Masculine**	**Feminine**
1st person	ወንድም_ኤ/ወንድሜ (my brother)		ወንድም_ኣችን/ወንድማችን （our brother）	
2nd person	ወንድም_ህ/ (your brother)	ወንድም-ሽ(your brother)	ወንድም _ኣችው/ወንድማችው （your brother）	
3rd person	ወንድም_ኡ (his brother）	ወንድም_ዋ(her brother）	ወንድም _ኣቸው/ወንድማቸው Thier brother	

Verb

A verb is a word that expresses an action, an occurrence or a state of being. Amharic verbs are very complex as in other Semitic languages [33]. A single Amharic verb may convey the subject and object of the sentence [35]. For instance, the word "ይነግረኛል" in English, it means 'he will tell me'; the subject (he) and the object (me) is implicitly stated in a single word. This verb "ይነግረኛል" is analyzed as follows. Verbal root: ngr ('to tell'). Verbal stem: negr ('will tell), subject: yi…al (he) and object: eñ (me).

Amharic verbs are consisting of a stem and ZERO or many affixes. Affixation can be prefix, infix, suffix and circumfix. The stem of a verb is composed of a root and a template. The roots represent the basic lexical component of the verb. It is characterized as a sequence of consonants or "radicals". The template consists of slots for vowels which are inserted among the consonants of a root to form a stem and it represents tense, aspect, mood, and one of a small set of derivational categories: passive-reflexive, transitive, causative, iterative, reciprocal, and causative reciprocal [35]. Amharic verbal stems consist of a root, vowels and template. For instance, sbr + ee + CVCVC forms the stem seber ('broke'). Each lexeme can appear in four different tense-aspect mood (TAM) categories. They are perfect (ive), imperfect (ive), jussive/imperative, and gerund (ive). Figure 2.1 [35] shows examples of Amharic verbs for each category.

- perfective: ደረሰ *der_es-e*
- imperfective: ይደርሳል *yI-ders-al*
- jussive: ይድረስ *yI-dres*
  imperative: ድረስ *dres*
- gerundive: ደርሶ *ders-o*

Figure 2.1: Examples of Amharic verbs in four TAM categories

Amharic verb must agree with its subject. Subject-Verb agreement is expressed by suffixes alone in some TAM categories (i.e., perfective and gerundive) and by a combination of prefixes and suffixes in other TAM categories (i.e., imperfective and jussive/ imperative). Baye Ymam [34] described these catagories of verbs as follow:

Perfective verbs are formed by adding suffixes like ኩ/ku, ክ/k, ሽ/sh, ኧ/e, አች/ch, አቹ/cu, ኡ/u that indicates person, gender and number to the perfect verb stem. For example, from a verb stem ሄድ ('hEd), perfective verbs such as ሄድኩ (hEdku), ሄድክ (hEdk) , ሄድሽ (hEdsh), ሄደ (hEde), ሄደች (hEdch), ሄዳቹ (hEdacu) and ሄዱ (hEdu) can be formed.

Gerundive verbs are formed by adding suffixes at the end of the gerundive verb to indicate person, gender and number. Suffixes such as: አሁ/ahu, አህ/ah, አሽ, አ/a, አች/ac, አን/an, አችሁ/chu and እተዋል/ewal can be added to a verb stem. For example, the stem "ሰር (sEr)": gerundive verbs such as ሰራሁ((sErahu),), ሰራህ(sErah), ሰራሽ(sErash), ሰራ(sEra), ሰራች(serac) ሰራን(sEran), ሰራችሁ(sEracu) and ሰርተዋል(sErtewal) can be formed.

Imperfective verbs are formed by affixing morphemes like ል-ኢ/l-i. ት-/t-, ት- ኢ/t-i, ይ- ኢ/y-i, ን - ኢ/ n-i, ት-ኡ/ t-u and ይ- ኡ/ y-u that indicates gender, person and number. The following are examples of imperfective verbs which are formed from the steam ሄድ (hEd): ልሂድ/lhId, ትሄድ/thEd, ትሄጂ/thEji, ይሂድ/yhId, ንሂድ/nhId, ትሄዱ /thEdu and ይሄዱ/yhEdu.

Jussive and imperative verbs are sometimes called mood and jussive verbs are used to express command for first and third persons whereas imperative verb is used to express the second person in the singular and plural form [34, 36].

Adjectives

Adjectives in a sentence modify nouns to denote quality of a thing which tells to what extent a thing is distinct from something else [33]. It comes before a noun to qualify a noun with some form of size, kind and behavior. For example, in the sentence "ጥቁር ቦርሳ (red bag)" the word "ጥቁር (red)" used to qualify the color of the noun "ቦርሳ (bag)". The morphology of Amharic adjectives is similar with Amharic noun.

Adverb

An adverb uses to qualify a verb by adding extra information to the sentence. Adverbs usually precede the verbs they modify or describe [34]. The followings are examples of Amharic adverbs: "ትናንት (Yesterday)", "ቶሎ/Quickly", "ከፉኛ/Hardly" and "አደገኛ/Dangerously".

Preposition

A preposition is a word that can be placed before a noun and perform adverbial operations related to place, time, cause and so on; which can't accept any suffix or prefix; and which is

never used to create a new word. Prepositions could not have any meaning alone but they will have meaning only when they are attached or used together with other words such as nouns and pronouns. Some of the prepositions include “ከ/from”, “ለ/to”, “ወደ/to”, “ስለ/for”, “እንደ/like”,” ጋር/with”, etc.

Conjunctions

Coordinate conjunctions are words that used to join two equal words, phrases, clauses and sentences. The most frequently used Amharic coordinate conjunctions are: “እና (and)”, “ግን (but)”, and “ወይም (or)”. For example, in the sentence, “ኢትዮጵያ የአሰብ እና የምፅዋ ወደብን መጠቀም ትችላለች። (Ethiopia can use both Aseb and Massawa Port.)” the conjunction “እና (and)” connects the two noun phrases “የአሰብ ወደብ (Aseb port)” and “የምፅዋ ወደብ/ (Massawa port)”.

2.1.3 Phrases

A phrase is a structure in a language that is constructed from one or more words. In Amharic, phrases are categorized into five categories: noun phrase, verb phrase, adjectival phrase, adverbial phrase and prepositional phrase [34,36]. Each phrase type can be categorized into “simple” (where only one word class is represented) and “complex” (where more than one word classes are represented).

Noun Phrases

A noun phrase (NP) is a syntactic unit in which the headword is a noun or a pronoun. The head of the phrase is always found at the end of the phrase. This type of phrase can be simple or complex. The simplest noun phrase consists of a single noun or pronoun such as: “እሱ (he)”, “እሷ (she)”, “እነሱ (they)”, etc. A complex noun phrase can consist of a noun (called the head) and other word classes such as complements, specifiers, adverbial and adjectival modifiers. These modifiers change the head from different aspects [34,36]. Table 2.3 shows examples of simple noun phrases of Amharic language which take suffixes indicating definiteness (DFF), accusative(ACC) case for direct objects, and prefixes representing prepositions.

Table 2.2: Amharic noun phrases

Noun Phrases	Amharic NP	English NP
Noun	ቦርሳ （borsa）	Bag
Noun + DEF	ቦርሳ-ው (borsaw)	The bag
Noun + DEF + ACC	ቦርሳ-ውን （borsawn）	The bag (as object of a verb)
PP + Noun + DWF	ለቦርሳ-ው (leborsaw)	To the bag

However, If the noun has a modifier, the modifier takes all these affixes. This situation is depicted in the table below.

Table 2.3: Amharic noun phrases with modifiers

Noun Phrases	Amharic NP	English NP
Adjective + Noun	ጥቁር ቦርሳ （tikur borsa）	Black bag
Adjective + DEF + Noun	ጥቁር-ኡ ቦርሳ (tikuru borsa)	The black bag
Adjective + DEF + ACC + Noun	ጥቁር--ኡን ቦርሳ （tikurun borsa）	The black bag (as object of a verb)
PP + Noun + DEF + Noun	ለጥቁር-ኡ ቦርሳ (letikuru borsa)	To black the bag

A complex noun phrase of Amharic language contains an embedded sentence within the phrase. For instance, "ማርታ የገዛችው ጥቁሩ ቦርሳ (the black bag that Martha bought)" is a complex NP whose head is "ቦርሳ /bag". This head is combined with the complement "ጥቁሩ/black" to produce the simple noun phrase "ጥቁሩ ቦርሳ (a black bag)". This noun phrase, in turn, has combined with the dependent clause "ማርታ የገዛችው (that Martha bought)" to produce the above complex NP. The presence of a preffix "የ" that attached to the verb indicates that the clause is a subordinate clause which cannot stand alone.

Verb Phrases

A verb phrase (VP) is composed of a verb as a head which is found at the end of the phrase, and other constituents such as complements, modifiers and specifiers [34,36]. Verb pharases also have simple and complex form as noun phrases. A complex verb phrase contains an embedded sentence that plays the role of a compliment or a modifier. Consider the following example, "በመኪና ወደ ትምህርት ቤት ሄደች (she went to school by car)". The adverb "በመኪና/by

car” and prepositional phrase “ወደ ትምህርት ቤት /to school” have modifyied the verb “ሄደች /went”.

Adjectival Phrases

The construction of Amharic adjectival phrase is similar to that of a noun phrase and a verb phrase. It is composed of an adjective (head), and other constituents such as complements, modifiers, and specifiers [34,36]. For example, “በጣም ትልቅ-ኡ /The very big”, “-ኡ /that” is a specifier, “በጣም/very” is a modifier modifying the head of the adjective “ትልቁ/big”.

Prepositional Phrase

A prepositional phrase is constructed from a preposition (head) and other constituents such as nouns, noun phrases, verbs, verb phrases, etc [34,36]. Unlike other types of phrases, the head of the phrase is found at the beginning of the prepositional phrases. For instance, in the prepositional phrase, “ከተማሪዎች ጋር ወደ ትምህርት ቤት /to school with students”, for instance, ከ _ጋር/ with” and “ወደ /to” are prepositions which are combined with the nouns “ተማሪዎች /students” and “ትምህርት ቤት /school”, respectively to form their prepositional phrase. The two prepositional phrases, in turn, combine to result in the bigger prepositional phrase that is provided in the example.

Adverbial Phrases

An Adverbial phrase is a phrase in which its headword is adverb [36]. It can be constructed from one or more adverbs. Adverbs refer to places, time or circumstances of the action mentioned by the verb. For example, “በፍጥነት መጣች/she came quickly”, በፍጥነት ‘quickly is the only adverb which describes the degree of the verb.

2.1.4 Clauses

A clause is a group of phrases. A clause should contain at list one verb phrase. There are two types of clauses: Independent clause (Main clause) and Dependent clause (subordinate clause).

Independent Clause (Main clause)

An independent clause is a clause that can stand alone and gives a meaningful and complete message. It is a simple sentence that contains a subject and a verb. For example, the clause: “ዩሃንስ ኳስ ገዛ/Yuhanis bought a ball”, has a subject, “ዩሃንስ/Yuhanis”; a verb, “ገዛ/bought”; and

an object, “ኳስ/ball”. The structure and types of Amharic simple sentence are discussed in the next section in more detail.

Dependent clause (subordinate clause)

A dependent (subordinate) clause is a clause that depends on another clause to make a complete sentence. Subordinate clauses contain both a subject and a verb, but do not express a complete thought.

In English, subordinate clauses can be detected by subordinate conjunctions. Some of English subordinate conjunctions are: “after”, “although”, “as”, “as if”, “because”, “before”, “even if”, “even though”, “if”, “in order to”, “since”, “though”, “unless”, “until”, “whatever”, “whether”, “when”, “whenever” and while. These types of clause act as an adverb in a sentence. Like an adverb, they modify a verb of the main clause by adding information that elaborates on when, where, why, how, how much or under what condition the action in the sentence takes place. Unlike English, Amharic subordinate clauses are recognizable by suffixes attached to the verb. For instance, the verb “ሰጠ/gave” can be changed to different forms by attaching affixes to it. (i.e., “ስለ-ሰጠ/because he gave”, “እየ-ሰጠ/while giving”, “ከ-ሰጠ/if he gave”, “ቢ-ሰጠ-እም/although he gave”, etc.).

A relative clause is a type of subordinate clause that acts as an adjective and describes a noun. In English, relative clauses contain a relative pronoun such as “who”, “whom”, “whose”, “that”, and “which”. On another hand, Amharic relative clauses can be identified by the morpheme “*የ/that*” which is attached to a verb.

Examples of Amharic relative clauses:

- **ትላንት የ-መጣው** *ሰውሥ the man* **who came yesterday**,
- **አበበ የ-አገኘውን** *ሰውሥ the man* **whom Abebe met**,
- **ሰላም የ-ገዛችው** *ጫማ/the shoes* **which/that Selam bought**,
- **ሴት ልጇ የ-ሞተባት** *ሴትሥ/the woman* **whose daughter died**,

2.1.5 Sentences

A sentence is a combination of one or more clauses that give a meaningful message. Unlike English which has a subject-verb-object sequence of words, Amharic language follows a subject-object-verb (SOV) grammatical pattern. For instance, the Amharic equivalent of the sentence "Martha went to school" is written as "ማርታ (Martha) ወደ (wede/to) ትምህርት ቤት (timehretbet/school) ሄደች (hedech/ went)". The principal constituent of Amharic sentences is subject and verb. The subject is a noun phrase that always precedes the verb of the sentence. It can be simple, defined, complex or compound. In Amharic sentence, a verb is always placed at the end of the sentence. Amharic sentences can be categorized into four based on their structure simple, compound, complex and complex-compound sentences [51].

Simple Sentence

A simple sentence is a sentence that contains a single independent clause. An independent clause is a group of words that has both a subject and a verb and expresses a complete thought. It is constructed from a simple noun phrase followed by a simple verb phrase that contains only one verb [51]. For example, "ማርታ መጽሃፍ ገዛች/ Martha bought a book" is an independent clause. It contains a subject ("ማርታ/Martha"), an object ("መጽሃፍ/book") and a verb ("ገዛች/ bought), and it expresses a complete thought. There are four types of simple sentences: Declarative, Interrogative, Imperative and Exclamatory.

Declarative sentences are used to state information. In Amharic, declarative sentences always end with the Amharic punctuation mark "፡፡" which is equivalent to a period (.) in English.

> **Example:** *"ሕብረት ኢንሹራንስ ካፒታሉን ወደ ግማሽ ቢሊዮን ብር አሳደገ፡፡/Hibret Insuraces has grown its capital to half billion Birr."*

Interrogative sentences are sentences that ask a question. In Amharic, these types of sentences always end with a question mark (i.e., "?").

> **Example**: "*ሕብረት ኢንሹራንስ ካፒታሉን ወደ ስንት አሳደገ? / by how much Hibret Insurance has grown its capital? "*

Imperative sentences mostly used to give a command or make a request and it ends with a period ("፡፡").

Example: *"መጽሃፉን ስጪኝ፡፡/give me the book."*

Exclamatory sentences express emotion. This type of sentence ends with an exclamatory mark (!).

Example: *ይገርማል! /amazing!*

Complex Sentence

A complex sentence has an independent clause joined with one or more subordinate clauses. Complex sentences contain more than one verb phrase [51]. For example, the sentence: "ወደ ውጭ የሚላከው የማር ምርት በከፍተኛ ሁኔታ እያሽቆለቆለ መምጣቱ ተገለጸ፡፡/ it is said that honey product that has been exported is significantly declining." is constructed from subordinate clauses which cannot stand alone. The noun phrase of the main clause contains a subordinate clause (i.e., "ወደ ውጭ የሚላከው የማር ምርት/ honey product that have been exported") which is formed from a simple noun phrase ("የማር ምርት/honey product") and a verb phrase (i.e., "ወደ ውጭ የሚላከው/the export of"). The verb phrase of the main clause also contains a subordinate clause (i.e., በከፍተኛ ሁኔታ እያሽቆለቆለ መምጣቱ / significantly declining"). It is formed from an adverbial phrase (i.e., "በከፍተኛ ሁኔታ/significantly") and a simple verb phrase ("እያሽቆለቆለ መምጣቱ/declining").

Compound Sentence

A compound sentence is a sentence that contains at least two independent clauses that have related ideas. The clauses are usually joined by a coordinator (coordinate conjunction). E.g., 'እና/and', 'ወይም/ or', 'ግን/but', and so on. Clauses that are joined by coordinate conjunctions are known as coordinate clauses. A coordinate clause is a part of a sentence that is equally important to the main clause. Some coordinate clauses give a complete message and can stand alone as a sentence but they might share the subject, the object, or the verb of the main clause.

Examples:

- አልማዝ ከሃረር እየመጣች ነው **ነገርግን** አበበ እቤት ውስጥ የለም።/Almaz is coming from Harar **but** Abebe is not in the house
- አበበ ወደ ሃረር ሊሄድ ይችላል **ወይም** አልማዝ ከሃረር ልትመጣ ትችላልች።/Abebe may go to Harar **or** Almaz may come from Harar.
- ማርታ ከትምህርት ቤት መጣች **እና** ተኛች።/Martha came home from school and slept.

Compound-Complex Sentences

A compound-complex sentence is basically formed by joining compound sentences and complex sentences. It contains at least two independent clauses and at least one subordinate clause. Since the independent clauses are coordinated by coordinating conjunction in a compound-complex sentence, they are called coordinate clauses. Here is an example of a compound-complex sentence, with the independent clauses highlighted in bold.

"የጠዋት ጸሃይ ስትወጣ (When **the sun sets*) የእግር ጉዞ ማድረግ እወዳለው (I like to take a walk)** *ነገር ግን (*but*) ዛሬ ዝናብ ስለነበረ (*since **it was raining today**) *እስከ ምሳ ሰሃት ድረስ ከቤት አልወጣውም (***I didn't get out from home until lunchtime***)."***

This sentence has two independent clauses and two dependent clauses. The dependent clause "የጠዋት ጸሃይ ስትወጣ (When the sun sets)" and "ዛሬ ዝናብ ስለነበረ (since it was raining today)" cannot stand on their own as a complete sentence; they are dependent. However, the independent clauses "የእግር ጉዞ ማድረግ እወዳለው (I like to take a walk))" and "እስከ ምሳ ሰሃት ድረስ ከቤት አልወጣውም (I didn't get out from home until lunchtime)" can be complete sentences on their own. The independent clauses are joined by coordinate conjunction "ነገር ግን (but)".

2.2 Open Information Extraction

2.2.1 Information Extraction

Information extraction (IE) is a field of computational linguistics that plays a crucial role in making information present in the raw text more accessible and organized for further processing [37]. The core task of IE is automatically extracting structured information from document collections. Typical IE subtasks include: Named Entity Recognition (NER), Co-reference or anaphora Resolution, Relation Extraction (RE) and Event Extraction (EE) tasks [4].

Named Entity Recognition (NER) is a task of recognizing and classifying mention of a named entity in the text. The task of NER begins by identifying proper names in free text, and then it classifies those entities into a set of predefined named entity types. Named entity types are specific to application but most IE systems search for common categories such as people, organizations, and places. The task of NER can be extended to identify and classify items that are not names or entities such as numeric values (e.g., measurements and prices), or expressions of times [33].

Co-reference or Anaphora Resolution is a task of finding all linguistic expressions that refer to the same entity. An entity can be referred to more than one time and in many different ways, in the same text. Co-reference or anaphora resolution is crucial for getting more accurate results in IE. There are several ways of referencing an entity [33]. An entity can be referenced by name-alias (e.g., “Addis Ababa University” and “AAU”), pronouns (e.g., “she”, “he”, “they”, and so on) and nominal (e.g., “Addis Ababa University” and “The University”).

Relation Extraction (RE) is a task of detecting and classifying relations that exist among the identified entities. Extracted relationships are represented by triples in the form <e1, rel, e2>, where e1 and e2 are entities, and rel is the relationship between the two entities [10].

Event Extraction (EE) is the task of identifying entities that play specific roles within an event referred to in the text [3]. An event can be defined as an incident happening at a given point of time and place. For example, given text documents containing information about some terrorist attacks, an IE system extracts event role filler, such as the place, the date, the number of people injured/killed, the organization, and etc. Therefore, the event extraction

system captures only important information of the entire story of a specific event and fills templates slots with a set of semantic roles for the typical entities involved in such an event rather than extracting disconnected facts and relations [3].

Romadhony *et al.* [10] classifies IE in to two broad catagories: the template-based and relation-discovery. Template-based IE extracts pre-specified class of entities, relationships, and events in natural language text. The information to be extracted is pre-specified in data objects containing information in per-defined and well-ordered structures called templates (or objects). Each template is consisting of a number of slots (or attributes). Thus, the extracted entities, relationships or events are fillers of the slots [39]. Relation-Discovery IE extracts structured information from text by discovering semantic relationships between entities and present them as a set of triplets {(e1, rel, e2}, where e1 and e2 are entities, and rel is the type of relationship relating the two entities. Unlike template-based, relation-discovery does not require to specify the type of information to be extracted prior to extraction; rather, it tries to find all types of relations in a given document.

2.2.2 A Brief History of Information Extraction

Although the revolution of IE emerged during the late 1970s, the first commercial IE systems appeared in the 1990s (e.g., JASPER [40]). In the 1990s, a series of annual workshops called the Message Understanding Conferences (MUC) [4] were initiated by the Defense Advanced Research Projects Agency (DARPA) which aimed to promote and evaluate the development of new methods in information extraction. The first conference (MUC-1) aimed to analyze naval operation messages. However, at that time, there was no standard format for recording information in the document and evaluating the performance of the systems. In MUC-2, the task was focused on automatic analysis of naval messages and extracting information based on pre-defined template slots. The MUC-2 template has 10 pre-defined slots to be filled by information extracted from naval messages and evaluation was done using primary evaluation measures (i.e., recall and precision). For MUC-3, the domain of interest had shifted to report terrorist events in Central and South America and the template extended to 18 slots. Figure 2.2 [41] shows an example of a template filling task from MUC-3 with 7 slots.

Sample Text

> "Three bombs have exploded in north-eastern Nigeria, killing 25 people and wounding 12 in an attack carried out by an Islamic sect. Authorities said the bombs exploded on Sunday afternoon in the city of Maiduguri.

Example templates

Incident type	Bombing
Time	Sunday afternoon
Location	Maiduguri
Perpetrator	Islamic sect
Dead-Count	25
Injured-Count	12
Instrument	Bomb

Figure 2.2: An extract of a MUC-3 template filling task

The task in MUC-4 is similar to MUC-3 but the number of slots has increased to 24 slots. MUC-5 was conducted as part of the Tipster program on international joint ventures and the electronic circuit fabrication domain. It had been focusing on two languages: English and Japanese. The template-filling task for MUC-6 involved the recognition of named entities (e.g., people and organizations), location names, temporal expressions, and numerical expressions. MUC-7 was focused on the identification of co-reference relations among noun phrases, the extraction of information about a specified class of events and the filling of a template for each instance of such an event.

Following MUCs, a similar series of programs have initiated. Automatic Content Extraction (ACE) and Text Analysis Conferences (TAC) are some of the programs which have encouraged progress in the field of IE [42]. The ACE series were held between 1999 and 2008, involving the detection of entities, relations, and events. Comparing to the task of MUC-7, the ACE series contains much more expressive relations of predefined types including the role of a person in an organization, part-whole relationships, location relationships, nearby locations, and social relationships. The TAC series was started following the ACE series. The main

focus was discovering relational information about named entities and then incorporate this information into a knowledge base which is known as Knowledge Base Population (KBP).

Most of the IE systems which participated in MUC, ACE and TAC programs require that the structures of information and domain of interest should be well defined prior to extraction. Thus, they cannot scale to massive and heterogeneous corpora like Web. In order to address the limitation of traditional template-based IE approaches, the relation-discovery approach was introduced in 2006. Shinyama and Sekine [43] presented the prior work in the relation-discovery approach described as "unrestricted relation discovery". Although the approach satisfies the important goal of avoiding relation specificity, it cannot scale to the Web because they used heavy linguistic processing which costs an O(D2) effort where D is the number of documents.

In recent years, discovering useful facts from a large and diverse corpus such as the Web has become a critical need. This need has triggered the introduction of a new extraction paradigm known as Open Information Extraction (OIE) [5]. Unlike traditional IE methods, OIE uses a relation-discovery approach, which facilitates the discovery of an unbounded number of relations from text and scales to large and diverse corpus such as the Web.

2.2.3 Overview of Open Information Extraction

The task of extracting information from the Web presents several challenges for traditional IE systems [44]. According to [45], the Web has several properties that make extracting information from the web is difficult for IE systems that employed traditional approaches. Firstly, The Web is heterogeneous. It contains all possible kinds of domains and article types, whereas most IE systems have concentrated on specific domains. Secondly, the relations of interest in the Web are often unanticipated, and their number can be large which also makes impractical to use an IE system that requires a predefined relation type. Lastly, the Web is massive which contains billions of documents, which means that a system would have to apply highly scalable extraction techniques.

Therefore, traditional IE is limited in terms of scalability and portability across domains and the performance of these systems is heavily dependent on considerable domain-specific knowledge [46]. These limitations of traditional IE systems have led to a paradigm shift toward the concept of OIE. Etzioni *et al.* [5] have introduced OIE, an information extraction

paradigm that extracts instances of an unrestricted set of relations, avoids domain-specific extraction, and scales linearly to handle Web-scale corpora [46]. There are three key properties that define an OIE system (i.e., automation, domain independence, and scalability) [44].

Automation

OIE systems aim to automate manual works require to build a distinct extractor for every relation of interest. Compared to traditional IE methods which model every relation of interest in a corpus separately, the cost to develop an OIE system is incurred once per language and it is independent of the number of target relations in a given language.

Domain-Independence

Due to the diversity of domains present in Web text, domain-independence is an important property of web IE systems. Although traditional IE systems perform well when trained and applied to a particular domain, they require the relations must be known and specified prior to extraction. Whereas OIE systems are designed to capture relational dependencies typically obtained via syntactic and semantic analysis using only domain-independent features that do not require deep linguistic processing.

Scalability

A Web IE system, which intended to process the immense and ever-growing number of web-pages, should be able to extract all relations of interest which are not known prior to extraction. However, Traditional IE systems process the entire corpus for every relation of interest which results in O (RD) system runtime that scales linearly for a collection of D documents and the R number of target relations. On another hand, OIE systems have the ability to extract information for all relations at once which results in O (D) runtime that scales linearly for D number of documents. Thus, OIE has a significant scalability advantage over traditional IE.

Table 2.4 [6] and Figure 2.3 [46] summarize the difference between traditional IE and OIE systems.

Table 2.4: Comparison of traditional IE and OIE.

	Traditional IE	**OIE**
Input	Corpus + Labelled Data	Corpus + Domain Independent Methods
Relations	Specified in advance	Free discovery
Development Cost	O(R), where R is number of relations	O (1)
Complexity	O (D * R), Where D is number of documents and R is number of relations	O (D), Where D is number of documents

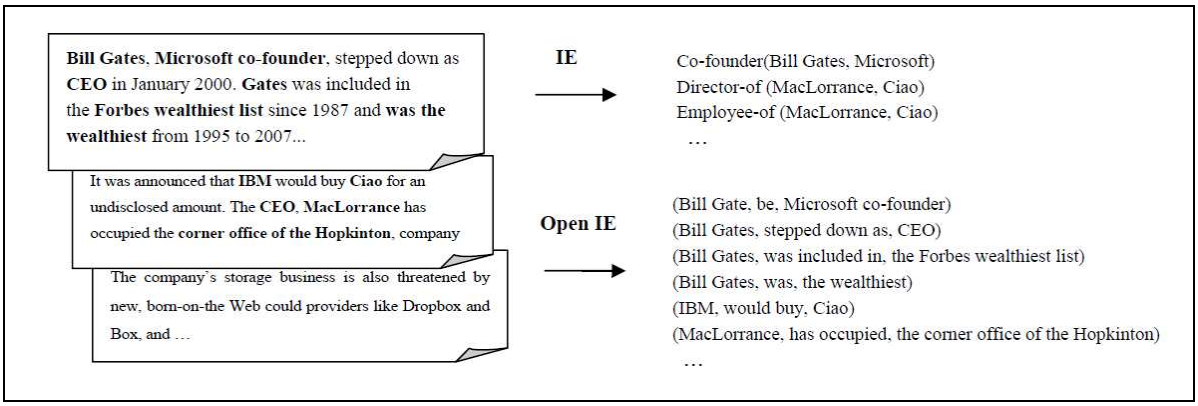

Figure 2.3: Comparison of traditional IE and OIE

2.2.4 Relation Categories in Open Information Extraction

OIE is the task of extracting arbitrary relations with their corresponding arguments from natural language text. By analyzing the nature of relations which can be extracted by most existing OIE systems, Bassa *et al.* [45] identified that all extracted relations fall into five categories. This section will give an overview of these categories.

Verb-based Relations

Verb-based relations are represented in predicate-argument structure as Rel (Arg1, Arg2), where, Rel is the verb of the sentence which express a relationship between the subject and the other arguments, Arg1 is the subject of the sentence and Arg2 is of the following types: indirect object, direct object, complement, or adverbial phrase. A well-formed sentence should

contain at least one verb. Thus, every common sentence contains at least one verb-based relation. According to [47], verb-based relations are most explicitly expressed relations in a sentence. Until OLLE [30], nearly all OIE systems extract only verb-based relations.

Noun-mediated Relation

Schmitz *et al.* [30] proposed to additionally cover relations mediated by other word classes other than verbs. A noun-mediated relation expresses the relations between two proper nouns in noun phrases. For example: "The runner of Ethiopia, Haile G. silasie", the noun "runner" relates "Haile G. silasie" and "Ethiopia". It can be represented in the predicate-argument structure as runner_of (Haile G. silasie, Ethiopia).

IS Relations

IS relations is an implicit relation because there is no explicit verb "is", but it can easily be deduced from the context. For example, from the phrase: "President Obama" we can extract a relation: IS (Obama, President). This kind of relation mostly found between two nouns.

HAS Relations

HAS relations are also expressed implicitly in a sentence. This kind of relations are mostly found in noun phrases. For example, from the phrase: "The son of Abebe", we can extract a relation: HAS (Abebe, Son).

Noun Compounds and Adjective Noun Pairs

Relations implicitly expressed in noun compounds and adjective-noun pairs are implemented in [59]. For example, from an adjective-noun pair: "red pen", a relation: (pen, that is, red) can be extracted and for the noun compounds: "plastic container" the relation (container, made of, plastic) can be extracted. However, the approach requires to obtain a proper synset from an external resource such as WordNet.

2.2.5 Outputs of Open Information Extraction

The output of an OIE system is a set of relations represented by different structures. Some systems use the binary structure to represent the extracted relations, while others used N-ary representation. This section discusses these representation mechanisms in more detail.

Binary Representation

Most of OIE systems extract binary relations and present them in three-component tuples either using predicate-argument structure in the form of {Rel (Arg1, Arg2)} or in the form of {(Arg1, Rel, Arg2)}, where Arg1 is the first entity, Arg2 is the second entity, and Rel is a predicate which represents the relationship between the two entities [37]. Consider, for instance, the following sentence:

"*Mulatu Astatke was born in the western Ethiopian city of Jimma in December 1943*."

An OIE system extracts the relationships between the entities and represent it as three-component tuples as follow:

(Mulatu Astatke, ***was born in****, the western Ethiopian city of Jimma)*
(Mulatu Astatke, ***was born in****, December 1943)*

Or represent in predicate-argument structure as follow:

E.g., ***was born in*** *(Mulatu Astatke, the western Ethiopian city of Jimma)*
was born in *(Mulatu Astatke, December 1943)*

N-ary Representation

The binary representation may lead to a critical information loss. For example, from the above sentence, "December 1943" and "the western Ethiopian city of Jimma" are found in separate relation tuple so that we could not get information about the relationship between the two phrases. N-ary is a better representation if more than one entity is related in a sentence. To generate N-ary representation of a relation instance, an argument for each of the remaining constituents preceding the relation phrase is created, in the order in which they appear. It has the following format: R = {Arg1, Rel, Arg2, Arg3, …, Argn} where Arg1 is the first entity, Rel is the relation phrase and Arg2, Arg3, Argn are other entities.

For instance, the relations in the sentence*:* "*Mulatu Astatke was born in the western Ethiopian city of Jimma in December 1943*." Can be represented in N-ary format as follow:

> {**Arg1:** *Mulatu Astatke,* **REL***: was born in,* **ARG2***: the western Ethiopian city of Jimma,* **ARG3:** *in December 1943*."}

2.3 Text Processing for Open Information Extraction

The task of an OIE system begins by processing the given document using several standard NLP procedures. The following subsections will introduce the most important NLP tasks used by OIE systems to process the input document.

2.3.1 Part-of-Speech Tagging

Part-of-Speech (POS) tagging is the task of labeling words with its most likely part of speech (e.g., noun, verb, adjective, adverbs, pronouns, determinants or articles, prepositions, numerals, conjunctions, particles, punctuation marks, etc.). A POS tagger takes a set of tags and a sequence of words as input and outputs a single best tag for each word. The process of POS tagging is dependent on the nature of the language. Different methods have been employed for the development of POS tagger such as rule-based, probabilistic, and hybrid of the two approaches. The rule-based method is a two-step process. First, a tagger assigns all possible parts-of-speech for each word without considering the context. Then, the tagger applies a large set of constraints to the input sentence to handle ambiguous and/or unknown words. The probabilistic method uses large labeled-corpus for training to compute probabilities of a word that occurs with a particular tag. Therefore, a tag will be assigned for a word based on probabilities assigned to each tag [48].

HaBit Amharic POS tagger is an Amharic POS tagging tool developed under the HaBit project [49]. The HaBit project aims to gather large-scale text data (corpora) from the web for under-resourced languages and to make shallow processing applications for these languages. To our knowledge, The HaBit Amharic POS tagger is the only publicly available Amharic POS tagging tool. It is developed by employing a TreeTagger trained on the corpus built by Walta Information Center (WIC) [50].

2.3.2 Phrasal Chunking

Phrase chunking, also known as shallow parsing, is the technique used to splits sentences into a set of non-overlapping phrases such as a noun or verb phrases aiming to simplify the sentence structure and identify entities. A chunker first identifies the chunk boundaries and then it labels chunks with their syntactic categories. It requires each token of the sentence to first be POS tagged. According to [48], there are two main methods to chunk a sentence: using regular expressions and training chunk parsers. The first method uses grammar based on regular expressions, which essentially defines rules using POS tag patterns. The second method uses a large corpus for training a chunk parser. The patterns that are learned rely on lexical items and/or POS tags as features.

Abeba Ibrahim [51] proposed an Amharic base phrase chunking and parsing which divides a sentence into different types of Amharic phrases by grouping syntactically correlated words which are found at a different level of the parser using Hidden Markov Model (HMM) model. Some rules are also used to correct some outputs of HMM-based chunker.

2.3.3 Dependency Parsing

Dependency parsing is a method for analyzing sentences into a directed graph that connects all words in the input based on a set of rules and criteria that define dependency relations between two words. Dependency relation is asymmetric and a binary relation, which means each word (i.e., head) is connected to exactly one other word (i.e., dependent) and the link will be labeled with the type of grammatical relation between the worlds (e.g., Subject, Predicate, Relative Clause, etc). The output of dependency parsing is a tree-structure that represents the sentence as a form of syntactic representation.

There are two methods of parsing a sentence: grammar-driven and data-driven. In grammar-based methods, a well-defined formal grammar is used to define the language either using context-free grammar (i.e., where dependencies are represented as production rules), or constraint satisfaction problems (i.e., where the analysis is restricted by a set of constraints present in the grammar). Whereas in data-driven methods, annotated data is used to learn probabilistic models of word-based dependencies [48].

Different language processing applications such as information extraction, question answering, machine translation and text summarization demand state-of-the-art dependency parsers. However, Amharic dependency parsing is a less researched area in Amharic NLP. Even though there are some initiatives (e.g., [52]) on creating linguistically annotated corpus such as Treebank which is important to train and develop an efficient dependency parser, Amharic dependency parser is not yet implemented.

2.3.4 Morphological Analysis

The term morphology in linguistics is defined as the study of the formulation of words and their internal structure. Morphological Analysis is the process of finding the morphemes of a word and providing grammatical information for the word based on the identified morphemes. A morpheme is a meaningful linguistic unit.

Morphological Analyzer is a computer program that takes words as input and identifies their stems and affixes and provides grammatical information about a word in a sentence [53]. For a morphologically complex language like Amharic, identifying morphemes of a word is very important for downstream NLP applications. For instance, morphological analysis has been used in OIE to extract patterns from sentences based on the grammatical information of words.

HornMorpho [53] is relatively complete morphological processing tool for Amharic and other languages spoken in Ethiopia (i.e., Tigrigna, Somali and Oromo). It is a rule-based program that analyzes Amharic words into their constituent morphemes (meaningful parts) and generates words, given root or stem and a representation of the word's grammatical structure.

2.3.5 Sentence Simplification

Since syntactically complex sentences often cause a challenge for most NLP systems, automatic sentence simplification has recently become an established research topic that aims to improve the performance of most NLP tasks such as parsing, machine translation, information extraction, and text summarization.

There are two main types of sentence simplification: lexical and syntactic. Lexical simplification focuses on replacing difficult, unfamiliar words of a sentence with more common terms and expressions. Syntactic simplification is a process of breaking down complex and long sentences into a set of simple sentences without losing essential information

[54]. Syntactic simplification is the most challenging task. It is a pipeline process comprising three stages: detection, splitting, and paraphrasing. The detection stage includes detecting boundaries of apposition, subordinate clauses, and coordinate clauses. During the splitting stage, the sentence will be split based on the detected boundaries. Finally, the paraphrasing stage converts all clauses into independent sentences by handling syntactic reordering and morphological changes.

Sentence simplification can be performed using manually crafted rules, statistical machine learning algorithms or a hybrid of the two approaches. It has been developed for English, Dutch, Swedish, French, and Portuguese. To the best of our knowledge, no work has been done on automatically simplifying Amharic sentences.

2.4 Approaches to Open Information Extraction

A typical OIE system takes natural language text as input, extract relation instances, and produces a set of extracted relations in different forms as output [39]. The core component of all OIE systems (i.e, extractor) uses generic patterns that express relations between entities using the grammatical structure of a sentence to extract relations. These patterns are either hand-crafted or learned from automatically labeled data [45]. The following subsection describes the two main approaches to OIE used for pattern creation, namely the knowledge-engineering approach and machine-learning approach.

2.4.1 Knowledge Engineering Approaches

Systems that employ knowledge engineering methods, make use of linguistic extraction patterns manually crafted by human experts through examining the nature of the language. Therefore, they require human experts to define rules or patterns for performing the extraction. Most of the relation extraction systems which participated in MUC utilized manually-crafted rules. A number of OIE systems (e.g., ReVerb [13], CORE [31], LSOE [56], DepOE [57], and ClausIE [58]) also used manually engineered rules operating on various levels of automatic linguistics analysis (i.e., shallow and deep syntactic parsing).

2.4.2 Learning-Based Approaches

Despite the fact that rule-based systems are simple, easier to interpret, and achieve higher results in terms of precision-recall and speed, the process of handcrafting rules is a very time consuming and ineffective task. It also requires a high level of expertise. To create such rules automatically, machine learning techniques have been used widely. There are three main categorizations of machine learning algorithms: supervised, semi-supervised and unsupervised.

Supervised method

Systems that employ a supervised method, learn a language model or a set of rules from a set of hand-tagged training documents. The training documents contain labeled text which represents the desired output of the model. The system takes a set of training documents as input and transforms the sentence into a set of features [59]. During the training process, the algorithm looks at both the features and the expected results and learns which features are the best predictors for certain results [60]. The training process continues until the model achieves a desired level of accuracy on the training data then applies the model or rules to a new text.

There are various types of supervised learning algorithms which have been applied to different subtasks of IE, such as Naïve Bayes, Support Vector Machines (SVM) [61], hidden Markov Models (HMM), Decision Tree, Sequential minimal optimization (SMO) and Conditional Random Fields [62].

Semi-supervised or weakly-supervised method

The major disadvantage of the supervised machine learning method is that manually developing a large tagged corpus requires a significant amount of labor and time. The semi-supervised or weakly-supervised method was introduced with the purpose to overcome the limitation of supervised approaches by reducing the amount of human supervision. This approach does not rely on annotated data; rather, it only requires either an existing ontology resource or a few seed instances as initial input to extract a large amount of information. Bootstrapping is the widely used weakly supervised method for relation extraction which learns extraction patterns form a large collection of documents and a few seed instances, and then it iteratively learns more relation instances and extraction patterns. Although it reduces

the amount of manual work required for preparing labeled data, bootstrapping has an error propagation issue. Error occurrence at the early training stages leads to more errors at later stages and decreases the accuracy of the extraction process [37]. Distant supervision [63] is the most recent method of the semi-supervised relation extraction method that combines the advantages of both bootstrapping and supervised learning. Distant supervision is a learning scheme that uses an already existing database to collect examples for the relation we want to extract.

Unsupervised Method

Both supervised and semi-supervised methods require manually defining the structures for the information to be extracted and annotating documents according to the defined structures. Therefore, to overcome this problem, recently, there has been an increasing amount of interest in unsupervised information extraction from large corpora. The goal of unsupervised relation extraction is to extract relations from the Web without labeled training data and without pre-specifying types of relations.

The Unsupervised method does not rely on the availability of labeled corpora for learning. Rather, it automatically labels its own training data. Systems that automatically label their own training data called self-supervised systems. The KnowItAll IE system [64] is an example of a self-supervised IE system that learns to label its own training examples using only a small set of domain-independent extraction patterns. It uses a set of generic patterns to automatically instantiate relation-specific extraction rules, and then learns domain-specific extraction rules and the whole process is repeated iteratively. However, the process requires a large number of search queries to be processed, thereby making it extremely slow. In order to reduce the amount of manual labor for crafting extraction rules and to prevent the unforeseen number of possible relations in the web corpus, self-supervised methods have been applied to some OIE systems TextRunner [5]) is one of self-supervised OIE system which employed self-supervised method to learn Naive Bayes Classifier.

2.5 Evaluating Open Information Extraction

Since there is no publicly available standard dataset for evaluation of the OIE task, different authors perform experiments on different datasets and use different metrics to report experiment results, such as precision and recall, F1 scores, and the area under the curve (AUC). As it is also infeasible to measure recall for a corpus as large as the Web, some authors reported precision only. Generally speaking, most of the works in OIE are evaluated based on human judges. Precision and recall of these systems are reported by judging each output of the OIE system as correct or incorrect according to a common criterion to all judges.

Precision

Precision of an OIE system shows the system's accuracy. It is the fraction of the number of returned correct extractions among the total number of returned extractions:

Precision = correct extractions /all returned extractions

Recall

Recall of an OIE system shows the coverage of the system and it is the fraction of returned correct extractions among all textual elements that are manually annotated.

Recall = Correct extractions/Manually Annotated Extractions

F1-score is the harmonic mean between precision and recall that measures an overall score of the system:

F1 = 2 * (Precision * Recall) / (Precision + Recall)

The precision-recall curve shows the tradeoff between precision and recall of different thresholds.

CHAPTER THERE: RELATED WORK

This chapter review related work in Open Information Extraction (OIE). A lot of OIE system are developed for different natural languages. Some of OIE systems which are developed for different natural language are reviewed below.

3.1 Open Information Extraction from English Text

Etzioni *et al*. [5] introduced TextRunner, an OIE system that trains a Naïve Bayes classifier with POS and NP-chunk features to extract relationships between entities. TextRunner consists of three modules: Self-Supervised Learner, Single-Pass Extractor and Redundancy-Based Assessor. Given a small unlabeled corpus and some relation-independent heuristics as input, the Self-Supervised Learner outputs a classifier that labels candidate extractions as "trustworthy" or not. The Single-Pass Extractor makes a single pass over the entire corpus and generates one or more candidate instances from each sentence, sends each candidate to the classifier, and retains the ones labeled as trustworthy. The Redundancy-Based Assessor assigns a probability to each retained instance based on a probabilistic model of redundancy in text. TextRunner was evaluated using a test corpus of 9 million Web documents and it obtained 7.8 million tuples. A set of 400 randomly selected tuples were evaluated by human reviewers and 80.4% were considered correct. Even though the system shows good results in extracting unrestricted relations, it extracts only explicitly expressed relations that are primarily word-based which means relations should occur between entity names within the same sentence. Subsequent work by the authors showed that employing the classifiers capable of modeling the sequential information inherited in the text, like linear chain CRF and Markov Logic Network can result in better extraction performance.

Another OIE system that is introduced by Wu and Weld [29] is called WOE. It uses Wikipedia as a source of training data. The WOE system generates training examples automatically by heuristically matching Wikipedia infobox values and corresponding text. The system has three main components: preprocessor, matcher and learner. The preprocessor converts raw Wikipedia text into a sequence of sentences. The matcher constructs training data from attribute-value pairs of infoboxes and matching sentences. Finally, the learner obtains the extraction patterns using either dependency path patterns or POS features. WOE can learn two

kinds of extractor: WOEparse learned from dependency path patterns, and WOEpos trained with shallow features like POS tags and NP chunks. Comparing with TextRunner, WOEpos runs at the same speed, but achieves an F-measure which is between 18% and 34% greater; WOEparse achieves an F-measure which is between 72% and 91% higher than that of TextRunner but runs about thirty times slower due to the time required for parsing. The authors concluded that dependency-parse features are highly informative when performing un-lexicalized extraction than shallow features.

Etzoni *et al.* [13] also introduced the successor of TextRunner called ReVerb, which aimed to prevent incoherent and uninformative extractions errors from TextRunner. To overcome these problems of TextRunner, the authors implemented the syntactic and lexical rules in the ReVerb OIE system. These rules serve two purposes: to eliminate incoherent extractions and to reduce uninformative extractions by capturing relations phrases expressed by a Verb-Noun combination that satisfies the pre-defined syntactic and lexical constraint such as light verb constructions (LVCs). Finally, ReVerb split the input sentence into an Argument-Verb-Argument triple. The authors reported that ReVerb achieved an AUC (area under Precision-Recall curve) twice as big as TextRunner and WOEpos, and 38% greater than WOEparse.

Aiming to improve OIE by covering a larger number of relation expressions and expanding OIE representation to allow additional context information such as attribution and clause modifiers, Mausam *et al.* [30] presented the system OLLIE. OLLIE first collects sentences from a corpus containing words that are part of a ReVerb triple. For each sentence, OLLIE computes the syntactic dependencies connecting the two relationship arguments and the relational word. Next, it annotates the relation node in the syntactic dependency path with the exact relation word and the PoS-tag, taken from the ReVerb triple associated with this sentence. By checking some constraints over the syntactic dependency tree, OLLIE generates extraction patterns. In the extraction phase, the system extracts the dependency path for any given sentence and matches it with one of the extraction patterns. Then, the associated extraction templates are used to identify the arguments of the relationship and the relational word. OLLIE found up to 146 times as many extractions for these relations than ReVerb and it obtained 1.9 to 2.7 times more area under precision yield curves compared to ReVerb.

OLLIE is superseded by RENOUN [65], which is a rule-based extractor incorporating most of the high precision learning of OLLIE. RENOUN builds a comprehensive list of relational nouns using bootstrapping over query logs and text. It then uses seed patterns to extract data and then uses these as a source of distant supervision for additional pattern learning.

Del Corro and Gemulla [58] presented a clause-based approach implemented in ClausIE. For each input sentence, ClausIE first computes the dependency parsing of the sentence and then determines the set of clauses using the dependency parsing. Next, for each clause, it determines the set of coherent derived clauses based on the dependency parsing and finally it generates propositions from the coherent clauses. They used hand-crafted rules utilizing the dependency structure of a sentence. According to the author, ClausIE achieved better precision than Reverb. ClauseIE's accuracy relies on the dependency parser used form parsing.

Xavier and Souza [56] proposed an OIE approach, named LSOE, where relations are obtained by matching lexical syntactic patterns. The extractor identifies relationships by applying lexical syntactic patterns and generic patterns that identify non-specified relationships based on the sentence structure. The authors reported that LSOE achieved better precision than ReVerb and DepOE.

3.2 Open Information Extraction for European languages

An OIE system for languages other than English is implemented in DepOE system by Gamallo *et al.* [57]. It used a rule-based analyzer and dependency parser to extract relations represented in English, Spanish, Portuguese, and Galician. DepOE first analyses each sentence of the input text using the dependency-based parser. For each parsed sentence, it discovers the verb clauses and the clause constituents, such as subject, direct object, attribute, and prepositional complements. Then, a set of rules is applied to the clause constituents to extract the target triples. The authors reported that for the same dataset accuracy of 68% was reached, while ReVerb reached 52% accuracy. The author concluded that the use of deep dependency parsing provides better accuracy and makes easier to build n-ary relations and to find important relationships that are not expressed by verbs.

3.3 Open Information Extraction for Asian Languages

Tseng *et al.* [31] presented a Chinese OIE so-called CORE. CORE has three modules: POS tagging, syntactic parsing, and entity-relation triple extraction. The authors adopted a Chinese text analyzer called CKIP for POS tagging and parsing, and the entity-relation triple is identified based on the labeling made in the syntactic parsing module. The authors reported that they achieved relatively promising F1 scores than Reverb. According to the authors, the evaluation results show that the CORE method is more suitable for Chinese open relation extraction than other RE methods.

Nam *et al.* [7] presented a Korean OIE system so-called SRDF. The SRDF system was designed to extract relation instances from Korean natural language text based on the use of singleton property and other NLP techniques such as part-of-speech tagging and chunking. SRDF also enabled extracting multiple numbers of triples from a single sentence via reification. The evaluation result was assessed by two human evaluators. The authors reported that the SRDF system archived 81% precision, 86% recall, and 83% F-score for detecting relation and 66% precision, 65% recall, and 65% F-score for generating triples.

Truong *et al.* [66] presented vnOIE, a system for Vietnamese OIE which took advantage of the grammatical clause-based approach. vnOIE exploited Vietnamese dependency paring to identify constituents of the input sentence (i.e., subject, verb, object, compliments, and adverb). After the constituents of the sentence are identified, vnOIE classifies the sentence into the predefined clause types. Finally, vnOIE will extract relations based on patterns of clause types. To generate a proposition as a relation triple (arg1, rel, arg2), the subject is considered as the first argument (arg1) of each clause; the verb, as a relation (rel). The remaining argument (arg2) such as an object, complement, and adverb will be determined by the connection with the relation. In their experiments, they evaluated the system using several factors such as grammatical structures of sentences and the number of verbs existing in a sentence. The authors reported that the average precision achieved by vnOIE is over 83% when clauses have four verbs or fewer. The results show that the lower the number of verbs in a sentence, the higher the precision. The system delivers the best result in the case of a one-verb clause, with a precision of 92.78%.

3.4 Information Extraction from Amharic Text

Getasew Tsedalu [32] developed an IE model for Amharic news text by making use of a supervised machine learning approach. It has four components: document preprocessing, text categorization, learning and extraction, and post-processing. The pre-processor handles the tokenization and normalization of the input document. The text categorizer is responsible for categorizing the news text into predefined categories. The learning and extraction component handles the extraction of candidate text and train the classifier model for predicting the category of the extracted candidate text. Finally, the postprocessor formats the extracted data and saves it to the database. The experiment was conducted on three different classification algorithms (i.e., Decision tree, Naïve Bayes and SMO). The experimental result showed that, among the three classification algorithms, the SMO algorithm performs better than others on both text categorization and IE.

Bekele Worku [33] developed a rule-based IE system for Amharic language text. It has three components: pre-processor, extractor, and post-processor. The pre-processor performs tokenization, sentence splitter, normalization, and stop word removal. The extraction component performs three tasks: (1) named entity recognition using rules and gazetteers that can identify the different named entity classes, (2) co-reference resolution using orthographic matching of strings and (3) extract surface-level relations that hold between the extracted named entities. Pre-defined relations with handcrafted rules are provided to the system. The last component (i.e., postprocessor) presents the extracted information to users using annotations. According to the authors, they obtained 90.5% Recall, 89.2% Precision and 89.8% F1-measure.

3.5 Summary

In summary, we identified two categories of OIE systems: data-based systems which make use of training data (i.e., TextRunner, WOE, and OLLIE) and rule-based systems which make use of hand-crafted rules or heuristics (i.e., CORE, ReVerb, LSOE, ClausIE, and DepOE). According to the experiment results, the rule-based systems achieved better accuracy than those based on classifiers trained by automatically generated training data.

Depending on the methods used for syntactic analysis, these systems can also be categorized into two: systems that used shallow syntactic analysis (i.e., chunking) and those used deep syntactic analysis (i.e., dependency parsing). While TextRunner, ReVerb, WOEpos, CORE, SRDF, and LSOE used POS tagging and shallow syntactic analysis, WOEparse, OLLIE, ClausIE, ReNoun, and DepOE made use of deep syntactic parsing. Experimental results showed that systems that are based on dependency analysis achieved significantly higher precision and recall than shallow features that relying on shallow syntax. However, the shallow feature-based approach is very promising in terms of speed, ease of implementation, and portability to other languages. Whereas deep syntactic parsing methods are prone to slow performance and their implementation is not easily available for many languages [30].

Moreover, in spite of their variety in method and aim, all OIE systems discussed above used language-dependent information for their implementation. Thus, the application of these systems poses challenges to languages that are different and morphologically complex like Amharic. The research works that are conducted on Amharic IE are very few in number and they contain unresolved issues. For instance, Getasew's work requires a large amount of manually annotated data for training. Selecting attributes for the classifier also requires the involvement of expert knowledge. Bekele's work also requires high human involvement since it requires hand-crafted extraction patterns. Besides, the input patterns are domain-dependent and they are not scalable to web text. Furthermore, both works need hand-crafted and domain-dependent templates. In consequence, an Amharic OIE which overcomes the limitations of the traditional Amharic IE has become a critical need. Thus, the aim of this research work is to deisgn OIE system for Amharic language.

CHAPTER FOUR: AMHARIC OPEN INFORMATION EXTRACTION

4.1 Introduction

In this chapter, we present Amharic Open Information Extraction (AOIE), the first Amharic OIE system, which is implemented based on a rule-based approach working on POS-tagged, morphologically analyzed and shallow parsed (chunked) sentences. The decision of using hand-crafted rules is due to high results in terms of precision-recall and speed. Although a rule-based approach operating on deep parsed sentences yields the most promising results for OIE systems, since there is no fully implemented dependency parser that is available for Amharic language, we have implemented a rule-based system operating on shallow parsed sentences. The general architecture and detail description of components of AOIE are discussed in the following sections.

4.2 System Architecture

The proposed Amharic OIE system (AOIE) has the following major components: Preprocessing, Morphological Analysis, Phrasal Chunking, Sentence Simplification, Relation Extraction, and Post-processing. AOIE accepts Amharic document as an input and outputs a set of relations in N-ary format. The preprocessor tags each word in the text with an appropriate POS tag and selects well-formed and informative sentences from the input text based on the POS tag of words of a sentence. The Morpological Analyser provides grammatical information of words of the input sentence. The phrasal chunker divides the input sentence into non-overlapping phrases based on POS and morphological tags of words. The sentence simplification component segments the sentence into a number of self-contained simple sentences that are easier to process. The relation extractor extracts relations from those simplified sentences and finally the post-processor prints extracted relations in N-ary format. The relationship between these components is shown in Figure 4.1.

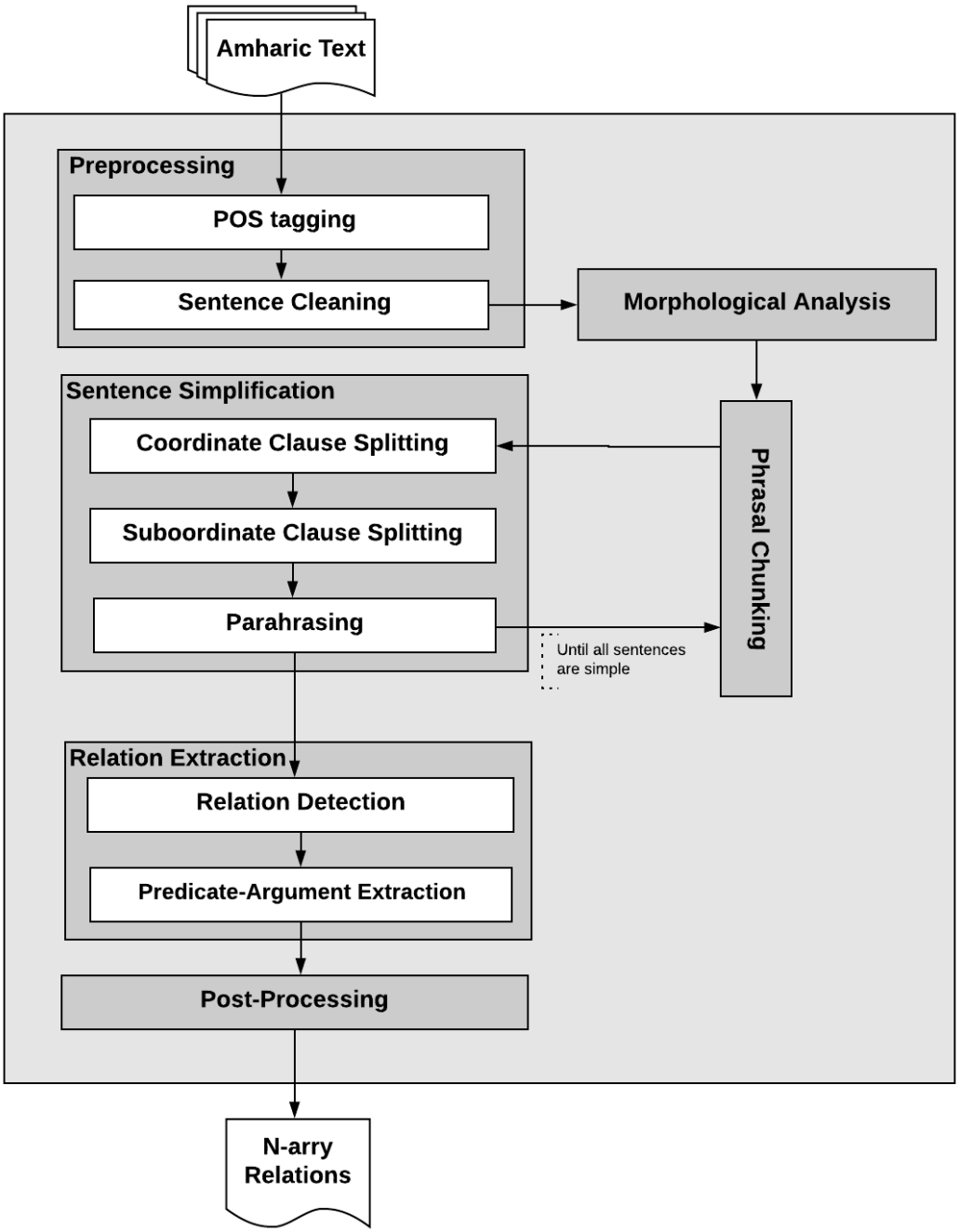

Figure 4.1: System architecture

4.3 Preprocessing

Preprocessing is the task that is used to make the input data ready for further processing. Two tasks are performed to make the data ready for subsequent processes: POS tagging and sentence cleaning.

4.3.1 POS Tagging

The preprocessing starts by tagging each word in the input text with their part-of-speech. Since HaBiT tagger is the only publicly available Amharic POS tagging tool, we have used HaBiT to obtain the part-of-speech tag of each word of the input text. Table 4.1 shows the tagset used in the HaBit Amharic tagger.

Table 4.1: POS tagset

POS tag	Definition of the tag
VN	Verbal/ infinitival Noun
NP	Noun attached with a preposition
NC	Noun attached with conjunction
NPC	Noun with a proclitic preposition and an enclitic conjunction
N	Any other noun
PRONP	Pronoun attached with preposition
PRONC	Pronoun attached with conjunction
PRONPC	Pronoun with a proclitic preposition and an enclitic conjunction
PRON	Any other Pronoun
AUX	Auxiliary verb
VREL	Relative verb
VP	Verb attached with preposition
VC	Verb attached with conjunction
VPC	Verb with a proclitic preposition and an enclitic conjunction
V	Any other Verb
ADJP	Adjective attached with preposition
ADJC	Adjective attached with conjunction
ADJPC	Adjective with a proclitic preposition and an enclitic conjunction
ADJ	Any other adjective
PREP	Preposition
CONJ	Conjunction
ADV	Adverb

NUMCR	Cardinal
NUMOR	Ordinal
NUMP	Numeral attached with preposition
NUMC	Numeral attached with conjunction
NUMPC	Numeral with a proclitic preposition and an enclitic conjunction
INT	Interjections
PUNC	Punctuation
UNC	Unclassified

4.3.2 Sentence Cleaning

After POS tagging each word of the input text, the sentence cleaning process splits the text into sentences and it filters out well-formed and informative sentences from the input text based on the POS tag of words. The punctuation marks such as period ('፡፡'), exclamatory mark ("!") and question mark ("?") are used as sentence delimiter to split the input text into sentences and the following conditions are checked to filter out well-formed and informative sentences.

i. The sentence should be declarative sentence because interrogative, exclamatory and imperative clauses don't contain any facts to extract as a relation.

e.g., "አሳ ሳንባ አለው?" doesn't contain any fact to be extracted.

ii. The sentence should have at least one verb at the end of the sentence. Otherwise, the sentences will be considered as malformed or uninformative.

iii. Information should not be extracted from direct speeches because they are likely a personal opinion that cannot be considered as fact.

e.g., አበበ "አሳ ሳንባ አለው" አለ፡

The sentence cleaning task automatically rejects sentences which don't satisfy the first and the second conditions, but for sentences containing direct speech, it assigns a special POS tag for the direct speech in quotation mark to prevent information extraction from direct speech. Non-declarative sentences can easily be detected by the punctuation mark found at the end (i.e.,'?','!'). The presence of a verb can be identified by examining the POS tag of the last word of a sentence and direct speeches can be recognized by quotation marks.

4.4 Morphological Analysis

The morphology and part-of-speech information of a word is very important to extract patterns from Amharic sentences. These patterns are used to formulate rules in order to apply them for phrasal chunking, sentence simplification, and relation extraction.

For morphological analysis, we utilized the most current version of HORNMORPHO (2.53) because it is relatively complete morphological processing tool for Amharic. Table 4.2 shows sample outputs of HORNMORPHO for the words: “ባካሄደው”, “ስብሰባው” “አድርጎ”, “ሾሟል”.

Table 4.2: Morphologically analyzed words

word: ባካሄደው **POS**: verb, root: <hyd>, **citation**: አካሄደ **subject**: 3, sing, masc **object**: 3, sing, masc **grammar**: perfective, iterative, transitive, relative, definite **preposition**: be	**word: አድርጎ** **POS**: verb, root: <drg>, **citation**: አደረገ **subject**: 3, sing, masc **grammar**: gerundive, transitive
word: ስብሰባው **POS:** noun, **stem**: ስብሰባ **grammar:** definite	**word: ሾሟል** **POS**: verb, root: <xwm>, **citation**: ሾመ **subject**: 3, sing, masc **grammar**: gerundive, aux:alle

Instead of tagging each word with its POS tag and morphological information separately, combining the POS tag and morphological information of a word, and use only one tag for a single word makes further processing easier. Thus, we created an additional 14 tags which hold both POS and morphological information. Table 4.3 lists custom tags that are created by combining POS and Morphology information.

Table 4.3: Custom created tags

Tags	POS	Morphology
ND	**N**	Definite or Accusive noun
NPD	**NP**	Definite or Accusive noun with prefixes
GV	**V**	Gerundive Verb
IV	**V**	Imperfective Verb
AV	**V**	Auxiliary Verb
PV	**V**	Perfective Verb
PAV	**V**	Passive Verb
PPV	**V**	Perfective Passive Verb
IRV	**V**	Imperfective Relative Verb
PRV	**V**	Perfective Relative Verb
PPRV	**V**	Perfective Relative Passive Verb
IRPV	**V**	Imperfective Relative Passive Verb
PGAV	**V**	Passive gerundive auxiliary verb
GAV	**V**	Gerundive Auxilary verb
INF	**N**	Infinitive

For example, the sentence "ዓባይ N ባንክ N ከ PREP 400 ADJ ሚሊዮን ADJ ብር N በላይ POSTP ትርፍ N ቢያስመዘግብም IV ዓለም N አቀፍ N ባንኮች N የወጣባቸው PRV ሕግ N ጫና ADJ እንዳሳደረበት PRV ገለጸ V ፡፡ PUNC" is tagged with both morphological and POS information. For instance, a POS tagger label the word "የወጣባቸው" as "VP" and a morphological analyzer return information which states the word are Perfective, Relative Verb. By combining the two information, the word "የወጣባቸው" is tagged with our custom tag PRV which indicates that the word is Perfective Relative Verb.

4.5 Phrasal Chunking

A Phrasal chunker divides a sentence into a set of non-overlapping phrasal chunks which are types of Noun Phrases (NP), Verb Phrases (VP), Prepositional Phrases (PP), adverbial Phrases (AP) and Adjectival Phrases (AJP). For instance, the sentence: "በተጠናቀቀው የጥቅምት ወር በኢትዮጵያ የምርት ገበያ የቡና ግብይት መጠን ከቀዳሚው ወር በ 74 በመቶ ጭማሪ የታየበት ሆኗል ፡፡" can be divided into phrasal chunks as follow "[በተጠናቀቀው የጥቅምት ወር] [በኢትዮጵያ የምርት ገበያ] [የቡና ግብይት መጠን] [ከቀዳሚው ወር] [በ 74 በመቶ ጭማሪ] [የታየበት ሆኗል] ፡፡"

Abeba Ibrahim [51] has implemented an Amharic phrasal chunker but its implementation is not available publicly. For that reason, we choose to design a new algorithm that simply chunks a given sentence at a higher level without labeling them with their type. The chunking algorithm accepts POS and morphological tagged Amharic sentences as input and generates chunks as outputs. The Algorithm processes the sentence from left to right. First, a chunk initiator is attached to the first token of the input sentence. By stopping at each token, it looking for a chunk terminator; if a chunk terminator is detected, the chunk will be ended their new chunk will be started, then a chunk initiator will be attached to the next token and it will be processed until the end of the sentence is detected. Algorithm 4.1 shows the implemented Amharic phrase chunking algorithm.

```
Input: POS and morphological tagged sentence (S)
Output: Amharic chunks (CHUNK_LIST)
Begin
 Initialize CHUNK_INITIAL to Zero.
 Initialize CHUNK_FINAL to position of the final token of S.
 For each token T in S
  If T is a chunk terminator
    Save the index of the token found before T into the varable
CHUNK_FINAL.
    Save the substring between CHUNK_INITIAL and CHUNK_FINAL into a
variable CHUNK.
    Add CHUNK to CHUNK_LIST.
    Save the index of next token to the varable CHUNK_INITIAL.
  End If
 End For
Output CHUNK_LIST
End
```

Algorithm 4.1: Phrasal chunking

A chunk terminator indicates a position where a chunk should end and a new chunk should begin. The POS and morphological tags assigned to every token are used to discover these positions. To mark a token in a sentence as a chunk terminator, the token should meet at least one of the conditions listed below. The seven conditions to be satisfied are elaborated below with examples. We used a symbol '**]**' to mark the end of a chunk, '**[**' to mark the beginning of a new chunk and the token which is being processed is written in bold.

Condition 1: Tag of the token should be CONJ (conjunction)

Example:

- [የአለም NP ልጅን ND] [ራሄል N] **እና CONJ** [ተስፋዬ N] [ከትምህርት NP ቤት N] [ትናንት ADV] አመጧት PV] ።

Condition 2: The tag of the token should be either NP (Noun with a preposition), PREP (preposition) or ADJ (Adjective) and the tag of the previous token should **not** be NP, PPRV (Perfective Passive Relative Verb) and PRV (Perfective Relative Verb).

Examples:

- [የአለም NP ልጅን ND] [ራሄል N] እና CONJ [ተስፋዬ N**] [ከትምህርት NP** ቤት N] [ትናንት ADV] አመጧት PV] ።
- [ፊፋ N] [የእግር NP ኳስ N ፌዴሬሽን N] **[አዲስ ADJ** አስመራጭ AN ኮሚቴ N] [እንዲመረጥ PAV አዘዘ N] ።

Condition 3: The tag of the token should be ND (Definite Noun) or ADV (Adverb) and the tag of the previous token should **not** be NP (Noun with Preposition).

Example:

- [የአለም NP ልጅን ND] [ራሄል N] እና CONJ [ተስፋዬ N] [ከትምህርት NP ቤት N] **[ትናንት ADV]** [አመጧት PV] ።

Condition 4: The tag of the token should be N (Noun) and the tag of the previous token should be ND (Definite Noun).

Example:

- [የአለም NP ልጅን ND] **[ራሄል** N] እና CONJ [ተስፋዬ N] [ከትምህርት NP ቤት N] [ትናንት ADV] [አመጧት PV] ።

Condition 5: The token should be a non-relative verb or ADJ (Adjective) and the next token should be the last non-relative verb.

Examples:

- [አለምአቀፍ ADJ] [የሩጫ NP ውድድር N] [በአዲስአበባ NP] **[ሊካሄድ** PAV ነው V] ።
- [ሚኒስትሩ ND] [ወታደሮቹ ND] [ሀገራቸውን ND [ከወራሪዎች NP ስለታደጉ PPRV] **[በጣም** ADJ] [አመሰገኑ PV] ።

Condition 6: The token should be the end of a sentence (i.e., "።").

Examples:

- [ሚኒስትሩ ND] [ወታደሮቹ ND] [ሀገራቸውን ND [ከወራሪዎች NP ስለታደጉ PPRV] [በጣም ADJ] [አመሰገኑ PV] ።
- [ሰራተኞቹ ND] [ድርጅቱ ND የሸለመውን PRV ኮከብ ADJ ሰራተኛ N] [ሊቀመንበር N] [አደረጉት PV] ።

Condition 7: The token should be the last non-relative verb and the previous token should **not** be a non-relative verb.

Examples

- [ሚኒስትሩ ND] [ወታደሮቹ ND] [ሀገራቸውን ND [ከወራሪዎች NP ስለታደጉ PPRV] [በጣም ADJ] **[አመሰገኑ PV]** ።
- [ሰራተኞቹ ND] [ድርጅቱ ND የሸለመውን PRV ኮከብ ADJ ሰራተኛ N] [ሊቀመንበር N] **[አደረጉት PV]** ።

4.6 Sentence Simplification

Generally, state-of-the-art OIE systems identify relationships between entities in a sentence by matching patterns over either its POS tags or its dependency tree. However, syntactically complex sentences where relevant relations often found in several clauses, possess a challenge for current OIE approaches which are prone to make incorrect and missing extractions.

To achieve higher accuracy on OIE tasks, we have implemented a sentence simplification algorithm that breaks down complex and compound sentences into simple sentences by applying a set of hand-crafted grammar rules. The rules are formulated based on various characterstics of Amharic sentences. In this way, sentences that have complex syntax are converted into a number of simple sentences that are easier to process without losing the original meaning.

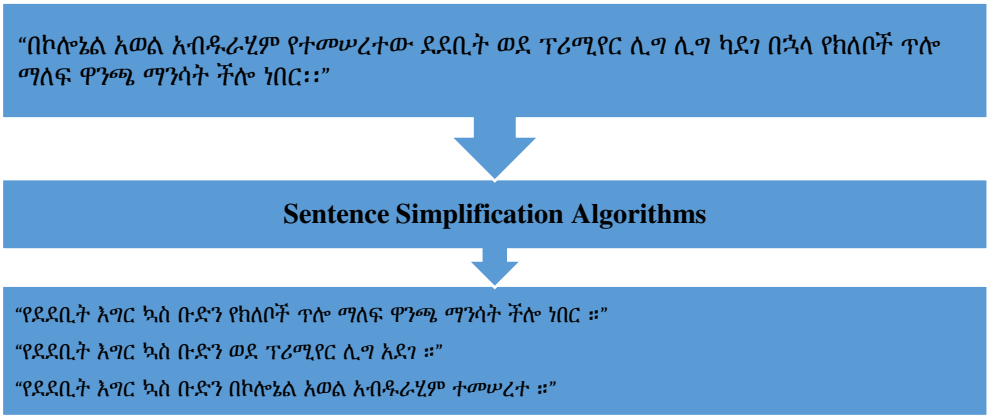

Figure 4.2: Input and outputs of sentence simplification algorithm

A set of simplification rules are used in order to split and/or paraphrase the sentences and generate new simple and independent sentences. Since resulting sentences might need further simplification, the process of syntactic simplification is structured in a recursive loop. The syntactic simplification loop starts by checking if the input sentence is a simple sentence by counting the number of chunks that contain a verb (verb phrases). If the sentence contains exactly one verb phrase, the sentence will be classified as a simple sentence. If more than one verb phrases which are joined by coordinate conjunctions are detected, the sentence will be classified as a compound sentence. Otherwise, the sentence will be classified as a complex sentence.

Here are examples of simple sentences which contain only one verb phrase:

- [ሚኒስትሩ ND] [ወታደሮቹ ND] **[አመሰገኑ PV]** ።
- [ሰራተኞቹ ND] [ኮከብ ADJ ሰራተኛውን ND] [ሊቀመንበር N] **[አደረጉት PV]** ።

Examples of compound sentences which contain more than one verb phrases which are joined by coordinate conjunction:

- [በኢትዮጵያ NP] እና CONJ [ኤርትራ NP] [መካከል ADV] [የአየር NP ትራንስፖርት N] **[ተጀምሯልPV] እንዲሁም CONJ** [ሁለቱም ADJ አገራት ND] [ኤምባሲዎቻቸውን ND] **[ከፍተዋል PV]**
- **[**አበበ N] [ወደ PREP ሃረር ND ሊሄድ IV] [**ይችላል** AV] **ወይም CONJ** [አልማዝ N] [ከሃረር NP **[ልትመጣ IV ትችላልች V]** ።

Examples of complex sentences that contain more than one verb phrases: verb phrases are highlighted in bold.

- [ሚኒስትሩ ND] [ወታደሮቹ ND] [ሀገራቸውን ND] **[ከወራሪዎች NP ስለታደጉ PPRV]** [በጣም ADJ] **[አመሰገኑ PV]** ።

- [ሰራተኞቹ ND] **[ድርጅቱ ND የሸለመውን PRV ኮከብ ADJ ሰራተኛ N]** [ሊቀመንበር N] **[አደረጉት PV]** ።

After classifying the sentences into simple, complex and compound, the sentences will be simplified based on its type. Algorithm 4.2 shows the implementation of the sentence simplification Algorithm. The simplification procedure is presented in the following sections in detail.

```
Input: POS and morphological tagged, and chunked sentence I
Output: Simple Sentences (SIMPLE_SENTENCE_LIST)
Begin
 Add I to SIMPLE_SENTENCE_LIST.
 IF all sentences in SIMPLE_SENTENCE_LIST are simple sentences
  Output SIMPLE_SENTENCE_LIST.
 End if
 FOR each sentence S in SIMPLE_SENTENCE_LIST.
  IF S is coordinate sentence
    Split Coordinate clauses and add to a variable CLAUSE_LIST.
  END IF
  IF S is complex sentence
    Split Subordinate clauses and add to a variable CLAUSE_LIST.
    Paraphrase the Clauses in CLAUSE_LIST and add clauses to
SIMPLE_SENTENCE_LIST.
  END IF
 END FOR
 Chunk all sentences found in SIMPLE_SENTENCE_LIST using Algorithm 4.1
 Repeat this algorithm for each sentence in SIMPLE_SENTENCE_LIST until we
obtain simple sentences.
End
```

Algorithm 4.2: Sentence simplification

4.6.1 Coordinate Clauses Splitting

In Amharic language, a compound sentence is composed of two or more independent clauses joined by a coordinating conjunction (i.e., "እና", "ወይም", "ግን", "ነገር ግን" "እንዲሁም", etc.). Semicolons and commas can also function as conjunctions. In order to split coordinate clauses in a compound sentence, we should first detect the coordinate conjunctions which joined the clauses and then we split the sentence at the conjunction. Coordinate conjunctions in a sentence are tagged as "CONJ" by the preprocessor so that they can easily be detected.

Before splitting the sentence, it is necessary to check both parts of the sentence are independent clauses. To be considered as an independent clause, they should at least have one verb. For instance, the sentence "[በኢትዮጵያ እና ኤርትራ መካከል የአየር ትራንስፖርት ተጀምሯል] clause1 ፤ [ሁለቱም አገራት ኤምባሲዎቻቸውን ከፍተዋል] clause2 ፤ [በዚሁ ሳምንትም ሁለቱ ሀገራት ዝግ ሆነ የቆየውን ድንበራቸውን ክፍት በማድረግ በየብስ ትራንስፖረት መገናኘት ጀምረዋል።] clause3" contains three complete and independent clauses which are joined by semicolons.

Algorithm 4.3 shows the implementation of a coordinate clause splitting algorithm. The algorithm accepts POS and morphologically tagged, and chunked sentences as input and returns a list of clauses. The algorithm first looks for coordinate conjunctions. If conjunction is detected, the sentence will be split there and the first part will be checked if it contains a verb. If it contains the verb, the first part will be added to the detected clause list and the remaining part of the sentence will be processed using a similar procedure.

```
Input: POS and morphological tagged, and chunked sentence (S)
Output: list of clauses (CLAUSE_LIST)
Begin
 Initialize INIT to Zero.
 For each token T in S which is tagged as "CONJ",
  Substring INIT to index of T and assign into a variable CLAUSE.
  Define a variable VERB_COUNT to the number of chunks in CLAUSE
which contain a verb type of (IV, AV, PV PAV, PPV, PGAV, GAV).
  Set INIT to the index of T.
  If VERB_COUNT >= 1
    Add CLAUSE to CLAUSE_LIST
  End If
 End For
Output CLAUSE_LIST
End
```

Algorithm 4.3: Coordinate clause splitting

4.6.2 Subordinate Clauses Splitting

A complex sentence has an independent clause joined with one or more subordinate clauses. Extracting embedded clauses from structurally complex sentences and generating new sentences from the extracted clause without affecting their original meaning, significantly reduces sentence length and complexity of the sentence.

Amharic subordinate clauses contain both a subject and a verb, but do not express a complete thought. In Amharic, subordinate clauses are recognizable by affixes attached to the verb. Here are some examples of verbs of Amharic subordinate clauses which are derived from the verb "መጣ": ቢመጣም (although he came), እንደመጣ (as he came), ስለመጣ (because he came), ከመጣ (if he came), ለመምጣት (in order to come, ካልመጣ (unless he came), እስኪመጣ (until he came), ሲመጣ (when he came) and እየመጣ (while he came). In addition, verbs of Amharic relative clauses have a prefix "Ye/የ" e.g., የመጣው (who/which/that came).

Based on the information obtained from the morphological analyzer, the preprocessor adds 'R' to the tag of the verb of the subordinate clause. Table 4.4 shows the types of verbs used in the subordinate clause.

Table 4.4: Verbs of subordinate clause

Tags	POS	Morphology	Examples
IRV	**V**	Imperfective Relative Verb	ስለሚመሰርተው, እንደማይመሰርት, ከሚመሰርት, ስለማይመሰርተው, የማይመሰርተው, እንደማይመሰርተው, ከማይመሰርተው, እንደሚመሰርተው, የሚመሰርተው
PRV	**V**	Perfective Relative Verb	ካልመሰረተው, እንደመሰረተ, ስለመሰረተ, ስለተመሰረተ, ስላልመሰረተው, የመሰረተው, እየመሰረተ, ካልመሰረተ, እንደመሰረተ, ከመሰረተ, ስለመሰረተ
PPRV	**V**	Perfective Relative Passive Verb	እንደተመሰረተ, ካልተመሰረተ, ስላልተመሰረተ, ስላልተመሰረተ ያልተመሰረተው, የተመሰረተው, የተመሰረተ, እየተመሰረተ, ካልተመሰረተ, ከተመሰረተ, ስለተመሰረተ
IRPV	**V**	Imperfective Relative Passive Verb	የማይመሰረት, የሚመሰረት እንደሚመሰረት, ስለሚመሰረት, የማይመሰረተው

For example, the sentence "[በኮሎኔል አወል አብዱራሂም የተመሠረተው PRV ደደቢት] [ወደ ፕሪሚየር ሊግ ሊግ ካደገ PRV በኋላ] [የክለቦች ጥሎ ማለፍ ዋንጫ] [ማንሳት ችሎ ነበር] ፡፡" has two subordinate clauses which are detected by the tag of verbs. The subordinate clauses are embedded in the noun phrase and verb phrase of the main clause (i.e., "በኮሎኔል አወል አብዱራሂም የተመሠረተው የደደቢት እግር ኳስ ቡድ" and "ወደ ፕሪሚየር ሊግ ካደገ").

As we can observe from the above example, the subordinate clauses can't stand alone rather they modify the subject, object or verb of the main clause. Therefore, in order to simplify the main clause, the subordinate clause splitting algorithm first separate the subordinate clauses from the main clause and then the subordinate clauses will be transformed into an independent sentence. Since the algorithm takes chunked sentences as input, the boundaries of chunks are used to identify the boundaries of the subordinate clause.

For example, from the above sentence, two subordinate clauses are detected.

- "ወደ ፕሪሚየር ሊግ ካደገ"
- "በኮሎኔል አወል አብዱራሂም የተመሠረተው የደደቢት እግር ኳስ ቡድን".

The boundaries of the clauses are the same as the boundary of the chunk that they located in. In order to produce well-formed sentences from the above clauses, all clauses (i.e., both main and subordinate) should be edited and paraphrased. Section 4.6.3 describes how paraphrasing is applied to clauses in more detail. After paraphrasing the above clause, the following three well-formed and independent sentences are generated.

- "የደደቢት እግር ኳስ ቡድን የክለቦች ጥሎ ማለፍ ዋንጫ ማንሳት ችሎ ነበር ።"
- "የደደቢት እግር ኳስ ቡድን ወደ ፕሪሚየር ሊግ አደገ ።"
- "የደደቢት እግር ኳስ ቡድን በኮሎኔል አወል አብዱራሂም ተመሠረተ ።".

Algorithm 4.4 shows the implementation of subordinate clause splitting algorithm. The algorithm accepts POS and morphologically tagged, and chunked sentence as input and return a simplified sentence. The algorithm iterates through all chunks of the sentence to look for a chunk contains a verb which is marked as 'R'. If found, the chunk will be marked as a subordinate clause and it will be removed from the main clause. Both resulting clauses (i.e., main clause and subordinate clauses) are paraphrased to form independent sentences.

```
Input: POS and morphological tagged, and chunked sentence S
Output: list of clauses (CLAUSE_LIST)
Begin
For each Chunk C in S
  If C contain a verb of type of (PRV, IRV, PPRV, IPRV)
     Add C to CLAUSE_LIST
     Remove C from S
  End If
End for
Add S to CLAUSE_LIST
Output CLAUSE_LIST
End
```

Algorithm 4.4: Subordinate clause splitting

4.6.3 Paraphrasing

Sentence paraphrasing is the process of rewriting a sentence while preserving its meaning. The process includes rearranging the order of a sentence in an SOV form and changing the verb form by removing prefixes of the verb. The paraphrasing task starts by checking if the clause is a relative clause or not by checking if the clause has a verb that is tagged as either IRV, IPRV, PPRV or PRV and has the morpheme "YE / የ". If the relative clause is detected, only the noun phrase found after the verb will remain and other words will be removed from the main clause and a new sentence will be generated from the relative clause.

For example, the sentence "[በቅርቡ NP የተቋቋመው PPRV የኢትዮጵያ NP አሠሪዎች AN ኮንፌዴሬሽን ND] [ከኢትዮጵያ NP ሠራተኞች N ማኅበራት N ኮንፌዴሬሽን N ጋር POSTP በሚያገናኟቸው IRV ጉዳዮች N ላይ POSTP] [አብሮ GV ለመሥራት NP] [የመግባቢያ NP ሰነድ N] [ተፈራረመ PV] ። PUNC" contains a relative clause "በቅርቡ NP የተቋቋመው PPRV የኢትዮጵያ NP አሠሪዎች AN ኮንፌዴሬሽን ND". After splitting the subordinate clause and paraphrasing the main clause, the main clause is reduced as "[የኢትዮጵያ NP አሠሪዎች AN ኮንፌዴሬሽን ND] [ከኢትዮጵያ NP ሠራተኞች N ማኅበራት N ኮንፌዴሬሽን N ጋር POSTP በሚያገናኟቸው IRV ጉዳዮች N ላይ POSTP] [አብሮ GV ለመሥራት NP] [የመግባቢያ NP ሰነድ N] [ተፈራረመ PV] ።" and new sentence (i.e.,"የኢትዮጵያ NP አሠሪዎች AN በቅርቡ NP ተቋቋመ PPV ። ") is generated from the subordinate clause (i.e., "በቅርቡ NP የተቋቋመው PPRV የኢትዮጵያ NP አሠሪዎች AN ኮንፌዴሬሽን ND").

To generate a simple sentence from the relative clause, the algorithm first checks if the voice of the verb is passive or active. If it is passive, the noun phrase found after the verb is a subject

and a noun phrase found before the verb is an object, and the morpheme "ye/የ" is removed from the verb. A new sentence is formed by combining the subject, object, and verb.

Example: "በኮሎኔል NP አወል N አብዱራሂም N **የተመሠረተው** PPRV ደደቢት N "

Subject: ደደቢት

Object: በኮሎኔል አወል አብዱራሂም

Verb: *ተመሠረተ*

Sentence: ደደቢት በኮሎኔል አወል አብዱራሂም *ተመሠረተ*።

If the verb has an active voice, the noun phrase found before the verb is a subject and a noun phrase found after a verb is an object and a postfix "ን" is added, and the prefix "ye/የ" is removed from the verb. A sentence will be formed by combining the subject, object, and verb.

Example: "ኮሎኔል NP አወል N አብዱራሂም N **የመሠረተው** PRV ደደቢት N "

Subject: ኮሎኔል NP አወል N አብዱራሂም

Object: ደደቢት + "ን"

Verb: *መሠረተ*

Sentence: ኮሎኔል አወል አብዱራሂም ደደቢትን *መሠረተ*።

However, the paraphrasing function only works for relative clauses that have perfective verbs (i.e., PRV and PPRV) because sentences produced from imperfective verbs are inconsistent in meaning with the original sentence. For example, consider the following two sentences.

- በየዓመቱ 500 ሚ. ብር የሚያወጣ IRV *መድሃኒት* ይቃጠላል ።
- የአዳማ ከተማን በእጥፍ **የሚያሳድግ** IRV ማስትር ፕላን ተዘጋጀ ።

From the subordinate clause of the first sentence "500 ሚ. ብር የሚያወጣ *መድሃኒት*" we can generate a sentence "*መድሃኒት* 500 ሚ ብር ያወጣል።" which implies that all medicine has a price of 500 million and from the second subordinate clause "የአዳማ ከተማን በእጥፍ **የሚያሳድግ** IRV ማስትር ፕላን", the sentence "ማስትር ፕላን የአዳማ ከተማን በእጥፍ ያሳድጋል።" can be generated. As we can observe, both newly generated sentences have different meaning than the original sentence. Therefore, for sentences which contain subordinate clauses with imperfective verb, we only reduce the original sentence by removing subordinate clause but we don't generate sentence from the subordinate clauses.

If other types of subordinate clauses (non-relative clauses) are detected, the whole clause will be removed from the main clause and a new sentence will be generated from the subordinate clause by removing the affixes of the verb of the subordinate clause. Unlike the relative clause, non-relative clauses might share the subject of the main clause. So, we have to merge the clause with all chunks found before the clause.

Example: the subordinate clause "ወደ ፕሪሚየር ሊግ ካደገ" is totally removed from the main clause and a sentence (i.e., "የደደቢት እግር ኳስ ቡድን ወደ ፕሪሚየር ሊግ አደገ") is formed.

Algorithm 4.5 shows the implementation of the paraphrasing algorithm. The algorithm takes a list of clauses as input, converts the input clause to well-formed sentences and outputs a list of sentences.

```
Input: List of POS and morphological tagged clauses (Cs )
Output: list of simple sentences (SENTENCE_LIST)
Begin
For each Clause C in Cs
 If C is relative clause
    Replace C in main clause by a noun phrase found after the verb
     If C contain  active verb
        Set SUBJECT by a noun phrase found before the verb
        Set OBJECT by a noun phrase found after the verb
     End If
     If C contain passive verb
        Set SUBJECT by a noun phrase found before the verb
        Set OBJECT by a noun phrase found after the verb
     End If
     Remove affixes from the verb
     Concatenate  SUBJECT,  OBJECT  and  verb  and  add  it  to
SENTENCE_LIST
  End If
  If C is not relative clause
     Remove affixes from the verb
     Concatenate C with all chunks found before C and add it to
SENTENCE_LIST
  End If
End For
  Add main clause to SENTENCE_LIST
  Output SENTENCE_LIST
End
```

Algorithm 4.5: Paraphrasing

Table 4.5 shows examples of complex and compound sentences which are broken down automatically by sentence simplification algorithm into less complex sentences.

Table 4.5: Examples of simplified sentences

Orginal Sentences	**Automaticly Processed Sentences**	
	Simplified Sentences	**Newly created Sentences**
[በእንግሊዝ NP የሚኖሩ IRV ኢትዮጵያውያን N] [የጋራ NP መድረክ N] [መሰረቱ V] ። PUNC	• ኢትዮጵያውያን N የጋራ NP መድረክ N መሰረቱ V ። PUNC	
[አረጋውያን ND] [ራሳቸውን ND የሚያቋቁሙበትን IRV ማህበር N] [መሰረቱ V] ። PUNC	• አረጋውያን ND ማህበር N መሰረቱ V ። PUNC	
[የጠዋት NP ጸሃይ N ስትወጣ IV] [የእግር NP ጉዞ N ማድረግ INF] [እወዳለው V] ነገር N ግን CONJ [ዛሬ ADV] [ዝናብ N ስለነበረ PRV] [እስከ PREP ምሳ N ሰሃት NP ድረስ POSTP] [ከቤት NP] [አልወጣሁም PV] ። PUNC	• የጠዋት NP ጸሃይ N ስትወጣ IV የእግር NP ጉዞ N ማድረግ INF እወዳለው V ። PUNC • ዛሬ ADV ምሳ N ሰሃት NP ድረስ POSTP ከቤት NP አልወጣሁም PV ። PUNC	• እስከ PREP ምሳ N ሰሃት NP ድረስ POSTP ዝናብ N ነበረ PPV ። PUNC
[በቅርቡ NP የተቋቋመው PPRV የኢትዮጵያ NP አሠሪዎች AN ኮንፌዴሬሽን ND] [ከኢትዮጵያ NP ሠራተኞች N ማኅበራት N ኮንፌዴሬሽን N ጋር POSTP በሚያገናኟቸው IRV ጉዳዮች N ላይ POSTP] [አብሮ GV ለመሥራት NP] [የመግባቢያ NP ሰነድ N] [ተፈራረመ PV] ። PUNC	የኢትዮጵያ NP አሠሪዎች AN ኮንፌዴሬሽን ND ከኢትዮጵያ NP ሠራተኞች N ማኅበራት N አብሮ GV ለመሥራት NP የመግባቢያ NP ሰነድ N ተፈራረመ PV ። PUNC	የኢትዮጵያ NP አሠሪዎች AN በቅርቡ NP ተቋቋመ PPV ። PUNC
[ለዓመታት NP በውጣ NP ውረዶች N የተፈተነው PPRV የኢትዮጵያ NP ትራንስፖርት N አሠሪዎች N ፌዴሬሽን] ND [ምሥረታ N] [ተከናወነ PAV] ። PUNC	የኢትዮጵያ NP ትራንስፖርት N አሠሪዎች N ፌዴሬሽን ND ምሥረታ N ተከናወነ PAV ። PUNC	የኢትዮጵያ NP ትራንስፖርት N አሠሪዎች N ለዓመታት NP በውጣ NP ውረዶች N ተፈተነ PPV ። PUNC

4.7 Relation Extraction

The main goal of an OIE system is to extract domain-independent and non-predefined relation instances from natural language text based on the grammatical structures of the language. In Section 2.2.4, five categories of relations that can be extracted from existing OIE systems are discussed in detail. In this section, we discuss how the five categories of relations can be detected and extracted from Amharic sentences.

4.7.1 Relation Detection

Verb-based Relations Detection

Since Amharic sentence has SOV structure, verb-based relations from Amharic sentence can be detected by the presence of a verb at the end of the sentence. Verb-based relations can be represented in a predicate-argument structure as Rel (Arg1, Arg2), where, Rel is the verb of the sentence, Arg1 is the subject of the sentence and Arg2 is of the following types: indirect object, direct object, complement, or adverbial. Correctly chunking the sentence into phrases is very important to identify the arguments of verb-based relations.

Examples:

- "አበበ ትላንትና ከአዲስ አበባ መጣ", the subject (i.e., "አበበ") and the adverbs (i.e., "አዲስ አበባ" and "ትላንትና") are related by the verb (i.e., "መጣ"). The extracted verb-based relations are:

  "መጣ" ("አበበ", "ከአዲስ አበባ")

  "መጣ" ("አበበ", "ትላንትና")

- "አበበ ፈጣን ነው", the verb (i.e., "ነው") expresses the relationship between the subject (i.e, "አበበ") and the compliment (i.e., "ፈጣን"). The extracted verb-based relation is:

  "ነው" ("አበበ", "ፈጣን")

- "አበበ ለከበደ መፅሃፍ ሰጠው", the verb (i.e., "ሰጠው") expresses the relationship between the subject (i.e., "አበበ") with the direct object (i.e., "ከበደ") and indirect object (i.e., "መፅሃፍ").

  "ሰጠው" ("አበበ", "ለከበደ")

  "ሰጠው" ("አበበ", "መፅሃፍ")

- “ትላንትና አበበ መፅሃፍ ገዛ”, the verb (i.e, “ገዛ”) express the relationship between the subject (i.e., “አበበ”) with the object (i.e., “መፅሃፍ”) and the adverb (i.e., “ትላንትና”).

  ገዛ (“አበበ”, “ትላንትና”)

  ገዛ (“አበበ”, “መፅሃፍ”)

HAS Relations

In Amharic sentences, HAS relation is implicitly expressed between two consecutive nouns or noun phrases when the morpheme “YE /የ” attached to the first noun or noun phrase.

Examples:

- “የአበበ ልጅ” ፡ HAS (“አበበ”, “ልጅ”)
- “የኢትዮጲያ ህዝብ”: HAS (“ኢትዮጲያ”, “ህዝብ”)
- “የአለም ጓደኛ”: HAS (“አለም”, “ጓደኛ”)
- የኢትዮጲያ NP የአየር NPሃይል: **HAS** (ኢትዮጲያ, አየር ሃይል)
- የኢትዮጲያ አየር መንገድ የህዝብ ግንኙነት ሃላፊ: **HAS** (ኢትዮጲያ አየር መንገድ. የህዝብ ግንኙነት ሃላፊ)
- የድሬዳዋ NP አስተዳደር N አዲስ ADJ ማስትር N ፕላን N : HAS(የድሬዳዋ አስተዳደር, አዲስ ማስትር ፕላን)

Noun-mediated Relation

In Amharic sentences, a common noun found between two proper nouns indicate the presence of a noun-medicated relation between two proper nouns. This mostly happens in appositions. Apposition is a grammatical construction in which two noun phrases are placed side by side with one element used to express the other differently.

Examples:

- “ኢትዮጲያዊው ዘፋኝ ጥላሁን ገሰሰ”: “**ዘፋኝ**” (“ጥላሁን ገሰሰ”, “ኢትዮጲያ”)
- “የአበበ ልጅ አቤል”: “**ልጅ**” (“አቤል, “አበበ”)
- “ራሺያዊው ጸሃፊ ፑሽኪን”: “**ጸሃፊ**” (“ፑሽኪን”, “ራሺያ”)

However, we used agent nouns instead of common nouns as an indicator of noun-mediated relations. Agent noun is a common noun that is derived from a verb denoting an action, and that identifies an entity that does the action. For example, “ተማሪ” is an agent noun formed from the verb “ተማረ". “አስፈጻሚ”, “መሪ”, “አሽከርካሪ”, “አስመጪ”, “አከፋፋይ”, “ላኪ” and “ዘፋኝ” are

some examples of agent nouns. The reason why we used agent nouns as an indicator of noun-mediated relation is that they are common nouns and the morphological analyzer can only able to distinguish agent nouns and identification of other common nouns requires knowledgebase.

For example: from a noun phrase" ኢትዮጵያዊው ዘፋኝ ጥላሁን ገሰሰ", the agent noun "ዘፋኝ" is used to detect a relation between "ኢትዮጵያ" and "ጥላሁን ገሰሰ" which indicate that "Tilahun Gesese is Ethiopian singer". It can be represented in a predicate-argument structure as "ዘፋኝ" ("ጥላሁን ገሰሰ", "ኢትዮጲያ").

IS Relation

IS relation is implicitly expressed relation between a proper noun and common noun.

Examples:

- ጣና ሃይቅ፡ IS (ጣና,ሃይቅ)
- ዳሸን ተራራ፡ IS (ዳሸን,ተራራ)
- ሜዲትራንያን ውቅያኖስ፡ IS (ሜዲትራንያን,ውቅያኖስ)

In Amharic, this kind of relation is found when a proper noun comes after a common noun. However, because of the same reason stated in noun-mediated relation detection, we only use agent nouns as a common noun to extract IS relation from noun phrases. See the following examples in which agent nouns are in bold and proper nouns are in italic and the relation extracted from the phrase is underlined.

Examples:

- የኢትዮጲያ **ሯጭ** *ሃይሌ ገብረስላሴ*: - **IS** (ሃይሌ ገብረስላሴ, **ሯጭ**)
- **ተማሪ** *ሰብለ*: - **IS** (ሰብለ, **ተማሪ**)
- *የብራዚሉ* **መሪ** *ጃይር ቦልሶናሮ*: **IS** (*ጃይር ቦልሶናሮ*, **መሪ**)

Noun Compounds and Adjective Noun Pairs

Different kinds of relation can be extracted from compound nouns. For example, from an adjective-noun pairs: "ቀይ ቀበሮ", a relation: *that_is* (ቀበሮ, "ቀይ") can be extracted and for the noun compounds: "የወርቅ ቀለበት" the relation: *made_of* (ቀለበት, ወርቅ) can be extracted.

However, relations implicitly expressed in noun compounds and adjective-noun pairs will not be covered in our AOIE system because they require external resources such as WordNet.

4.7.2 Predicate-Argument Extraction

There are altogether four categories of relations (i.e., Verb-based, HAS, IS and Noun-mediated relations) can be extracted based on the grammatical structure of Amharic sentence. This section presents all algorithms which are implemented for each category of relations.

Verb- based Relations

The phrasal chunking offers an easy way to extract a verb-based relation from simple sentences. A verb phrase is often found at the end of a sentence, the subject is mostly found at the beginning of the sentence. For example, consider the following chunked sentences.

- [የደደቢት NP እግር N ኳስ N ቡድን N] [በኮሎኔል N አወል N አብዱራሂም N] [ተመሠረተ PAV]
- [ኮሎኔል N አወል N አብዱራሂም N] [ደደቢትን ND] [በ1989 ዓ. ም .] [መሠረተ V]

The verb-based relation extraction algorithm is shown in Algorithm 4.6. The algorithm excutes the following three action sequences.

1. It searchs for a verb-containing phrase in a sentence; if detected, it marks the phrase as relation phrase.

   Sentence 1: Relation phrase = "ተመሠረተ"

   Sentence 2: Relation phrase = "መሠረተ"

2. It searchs for a noun phrase at the beginning of the sentence; if detected, it marks the noun phrase as the first argument and adds all remaining phrases to the argument list.

   Sentence 1: Argument1= "የደደቢት እግር ኳስ ቡድን"

   Argument List = {"በኮሎኔል አወል አብዱራሂም"}

   Sentence 2: Argument1 = "ኮሎኔል አወል አብዱራሂም"

   Argument List = {"ደደቢትን", "በ1989 ዓ. ም ." }

3. It returns the component in the form of a set of predicate-argument structure: Rel (Arg1, Arg2) where Rel is a phrase which was marked as a relation phrase, Arg1 is a

phrase which was marked as first argument and Arg2 is a phrase found in the argument list.

Sentence 1: **መሠረተ** (“ኮሎኔል አወል አብዱራሂም”, “ደደቢትን”)

Sentence 1: **መሠረተ** (“ኮሎኔል አወል አብዱራሂም”, “በ1989 ዓ. ም .”)

Sentence 2: **ተመሠረተ** (“የደደቢት እግር ኳስ ቡድን”, “በኮሎኔል አወል አብዱራሂም”)

```
INPUT: POS and morphological tagged, and chunked sentence (S)
OUTPUT: relation tuples in predicate argument structure
BEGIN
   For each chunk C in S.
        IF C contain a verb
           Predicate = C
        End IF
        IF C is the first chunk and it is a noun phrase
           Argument1 = C
        END IF
        IF C does not contain a verb
          Add C to ArgumentList
        END IF
    END FOR
    output the relation in form of predicate (Argument1,
ArgumentList))
END
```

Algorithm 4.6: Verb-based relation extraction

HAS Relation

HAS relation expresses a binary relation between X and Y, where X and Y are noun or noun phrases. Algorithm 4.7 shows the implementation of the HAS relation extraction algorithm. The algorithm takes POS and morphologically tagged noun phrases. It checks if a noun with a morpheme “YE/የ” found in the input noun phrase. If found, the word itself is the first argument and words found after it is the second argument.

Examples:

የአዲስአበባ NP የታክሲ NP አሽከርካሪዎች N: **HAS** (አዲስአበባ, የታክሲ አሽከርካሪዎች)

የብራዚሉ N መሪ AN: **HAS** (ብራዚል, መሪ)

```
OUTPUT: HAS relation tuple in predicate argument structure
BEGIN
   IF a noun, which has a morpheme "YE/የ", followed by other noun
is found
        Argument1 = the noun itself
        Argument2 = the next noun
  END IF
 output the predicate in form of HAS (Argument1, Argument2))
END
```

Algorithm 4.7: HAS relation extraction

IS Relation

IS relation extraction algorithm captures binary relations when a common noun present right before a proper noun.

Examples:

- ተማሪ AN አምሃ N: **IS** (አምሃ,ተማሪ)
- የታክሲው NP አሽከርካሪ AN አቶ N ሃይሉ N: **IS** (አቶ ሃይሉ, አሽከርካሪ)
- የብራዚሉ NP መሪ AN ጃይር N ቦልሶናሮ N: **IS** (ጃይር ቦልሶናሮ, መሪ)

The implementation of IS relation extraction is shown in Algorithm 4.8. The algorithm takes POS and morphologically tagged noun phrases as input. By iterating to each word, it looks for an agent noun which is followed by another noun. If found, the words found after the agent noun is extracted as the first argument and the agent noun will be extracted as the second argument.

```
INPUT: POS and morphological tagged Noun Phrase (NP)
OUTPUT: IS-A relation tuple in predicate argument structure
BEGIN
  If an agent noun followed by a noun is found
        Argument1 = noun phrase found after the agent noun
        Argument2 = the agent noun
  END If
  output the predicate in form of IS (Argument1, Argument2))
END
```

Algorithm 4.8: IS relation extraction

Noun-Mediated Relation Extraction

Noun-mediated relation expresses a binary relation between two nouns. Agent nouns are used to detect noun-mediated relations. The implementation of noun-mediated relation extraction is demonstrated in Algorithm 4.9. The algorithm takes POS and morphologically tagged noun phrases as input. The algorithm first looks for an agent noun that is found between two nouns. If found, the noun found after the agent noun is extracted as the first argument, the noun found before the agent noun is extracted as the second argument, and the agent noun will be extracted as the predicate.

```
INPUT: POS and morphological tagged Noun phrase (NP)
OUTPUT: noun-medicated relation tuple in Predicate-Argument
structure
BEGIN
  IF an agent noun is found between two nouns
       Predicate = agent noun
       Argument1 = noun phrase found after the agent noun
       Argument2 = noun phrase found before the agent noun
  END IF
output the predicate in form of predicate (Argument1, Argument2)
END
```

Algorithm 4.9: Noun-mediated relation extraction

4.8 Post-processing

The post-processing component used to present the outputs of the predicate-argument extractor in the form of N-ary representation. The separation of the extraction and representation of relations is a recommended approach because the representation can easily be changed without affecting the extraction. For example, the following two relations are obtained from verb-based relation extractor:

- R1 = "መጣ" ("አበበ", "ከአዲስ አበባ").
- R2 = "መጣ" ("አበበ", "ትላንትና").

The extracted relations are binary relations that show a relation between two entities. However, the binary representation often leads to a critical information loss. For example, in R1 and R2, the separation of the two constituents: "ከአዲስ አበባ") and "ትላንትና") leads to the loss of information. As a result of this, we used N-ary representation to represent the extracted relations. To generate an N-ary representation of a relation tuple, an argument for each entity

will be created in the order in which they appear. It has the following format: Rel (Arg1, Arg2, Arg3, …, Argn). For example, the N-ary representation of the extracted relation is:

- “መጣ” (Arg1: “አበበ”, Arg2: “ከአዲስ አበባ”, Arg3: “ትላንትና”)

4.9 Prototype

The proposed method and algorithms are implemented in prototype software. The prototype software is developed under an object-oriented environment, using Microsoft C# as a programming language. The prototype takes a POS-tagged text file in UTP-8 format as input, performs sentence-by-sentence processing and produces a set of relations in N-ary format.

The Morphological Analysis component analyses the morphology of each word of the input sentence using HORNMORPHO and it assigns custom tags to the word. Then, the Phrasal Chunking component chunks the sentence into a set of phrases. If the sentence is compound or complex, the sentence will be passed to Sentence Simplification component that simplifies the sentence and passes the sentence to the relation extracton component. The Relation Extraction component extracts a set of relation from the input sentence. Finally, the Postprocessor displays the extracted relations in N-ary format. Figrue 4.3 shows user interface of the Prototype.

The input of the prototype is a POS-tagged text. It can be opened from the location where it is stored as a file by using the “Browse” button. By pressing “Morphological Tagging” button, the input text will be splitted into a set of sentences and words of each sentences will be morphologically analized. When the “Chunk” button is pressed, each sentence will be chunked into a set of phrases. By pressing “Simplifiy” button, complex and compound sentences will be segmented into a set of simple sentences. Finally, relations will be extracted from input sentences and the results will be displayed by pressing “Extract Relation” button.

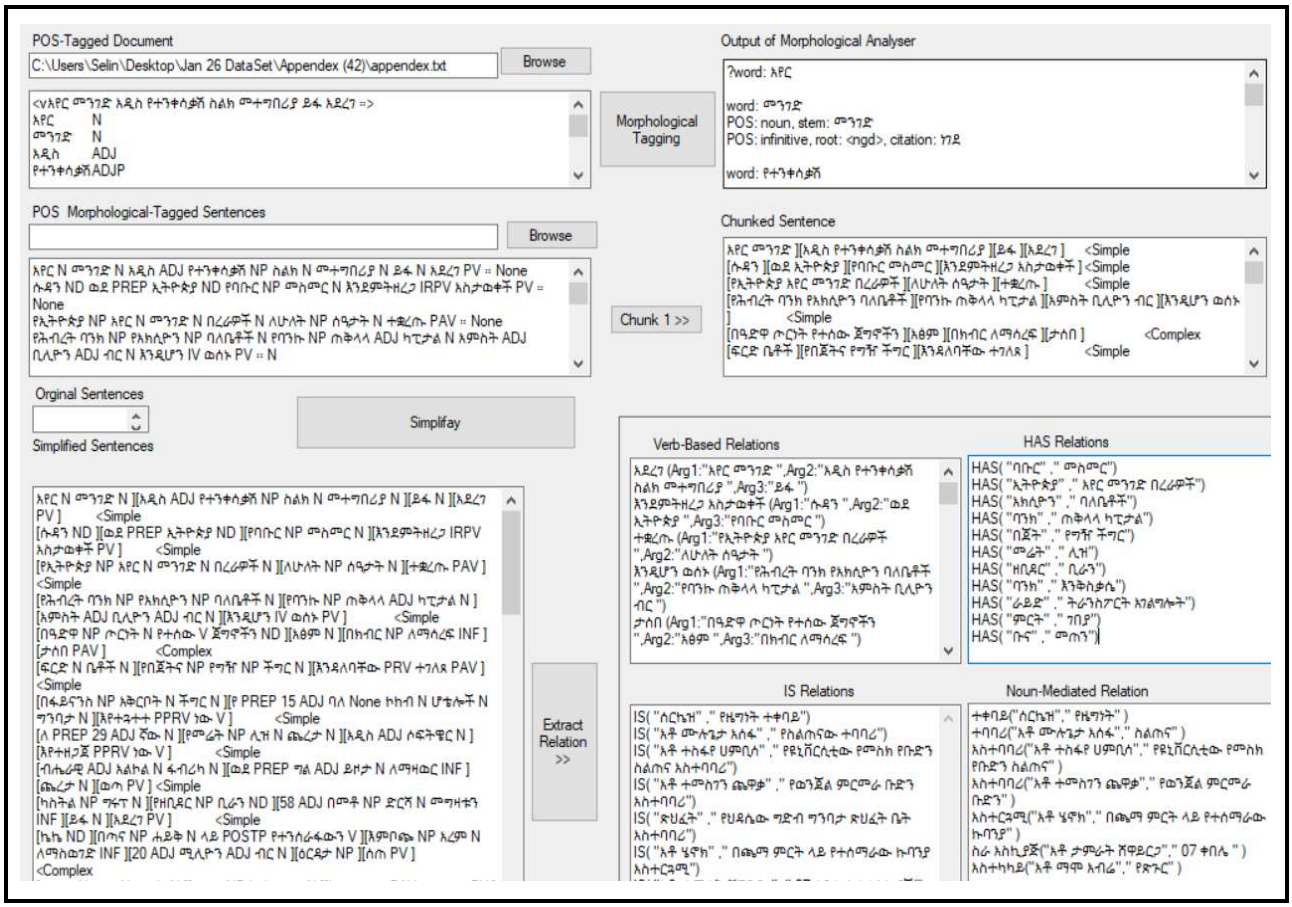

Figure 4.3: User interface of the Prototype

CHAPTER FIVE: EXPERIMENT

This chapter describes the experiment conducted to evaluate the performance of the proposed method and algorithms. It explains how the test datasets were created, and how the sentences were annotated. Furthermore, the results of the evaluation are discussed.

5.1 Experiment Setup

In order to reduce evaluation time and increase the efficiency of the evaluation process, we have implemented an evaluator which automatically compares results of AOIE system to the manually annotated dataset and calculates precision. The comparison function ignores minor differences like punctuation marks or other special characters since they don't change the meaning of extracted information. The order of arguments found after the relation phrase is also ignored.

5.2 Datasets

The dataset used in the experiment has 215 POS-tagged sentences consisting of 2482 words. The sentences were collected from online Amharic news sources such as the Reporter Ethiopia [67] and Walta Media and Communication Corporate [68]. The sentences were collected randomly from different domain areas to study the domain-sensitivity of AOIE system. We used HABIT Amharic tagger [14] to label each word with its POS tag and HornMorpho [16] for morphological analysis. Regarding the complexity of the sentences, the dataset consists of 75 simple sentences that contain exactly one verb phrase, 119 complex sentences that have more than one verb phrase and 21 complex compound sentences. A detailed histogram is shown in Figure 5.1.

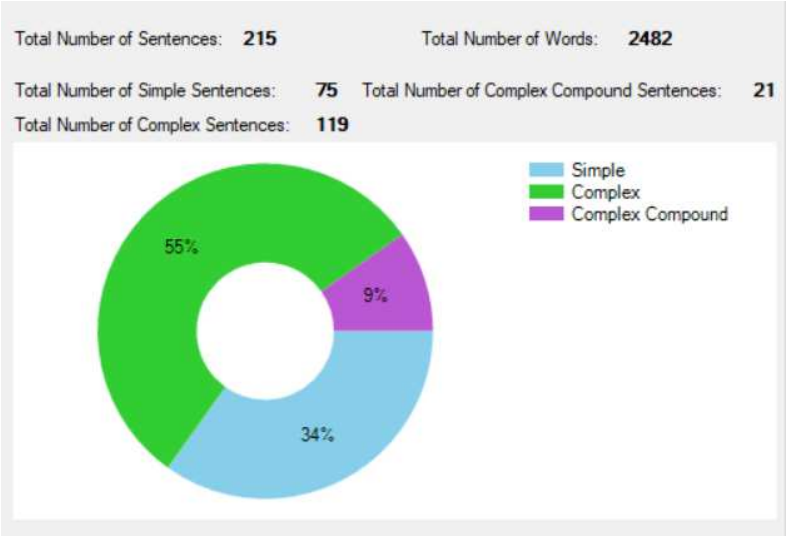

Figure 5.1: Histogram of dataset

5.3 Evaluation Results

We have evaluated the sentence simplification algorithm separately because the performance of the sentence simplification algorithm has a significant effect on the performance of the extraction system.

5.3.1 Evaluation Result of Sentence Simplification

The sentence simplification algorithm is tested using 130 complex and compound sentences. From 130 sentences, the algorithm generated 229 simple sentences. The researcher evaluated each sentence as correct or incorrect. The result of the evaluation showed that, out of 229 clauses, 74% of embedded clauses were correctly identified and 72% correctly paraphrased. Table 5.1 summarizes the findings.

Table 5.1: Evaluation result of sentence simpliciation

	Total	**Correct**	**Precision**
Clause identification	99	74	0.74
Parahrasing	229	167	0.72
Overall Algorithm			**0.73**

5.3.2 Evaluation Result of Extraction

In order to create the annotated dataset for extraction, each sentence had been tagged by the researcher with all possible relation tuples in the form of [REL] ([Arg1], [Arg2], [Arg3] [Argn]). We annotated 768 relations manually. All sentences at least have one distinct relation tuple extracted from it. The annotated dataset consists of 414 verb-based relations, 207 HAS relations, 78 noun-mediated relations and 69 IS relations.

The automatic evaluator labels extracted relations that match manually annotated items as "correct" and relations that don't match manually annotated items are labeled as "incorrect". Thus, all returned extractions are the sum of items that have been extracted correctly and incorrectly. To evaluate extraction algorithms of AOIE, we have used precision. Table 5.2 shows the result of the evaluation. A total of 556 correct relations tuples are extracted from the dataset.

Table 5.2: Evaluation result of relation extraction

	Manually Annotated Sentences	**Automatically Extracted Sentences**	**Correct Extracted Sentences**	**Precision**
Verb-Based	414	310	286	**0.92**
HAS	207	140	128	**0.91**
IS	69	50	39	**0.78**
Noun-Mediated	78	56	41	**0.73**
TOTAL	**768**	**556**	**494**	**0.88**

5.4 Discussion

In this section, we discuss the evaluation results in detail. As we can observe from Table 5.2, the AOIE system has extracted 556 instances of relation from 215 sentences with a precision of 88%. Considering the errors made by AOIE, in general, we found that 56.5% of errors are due to complex and malformed sentences. Although the performance of AOIE can be significantly improved by simplifying complex sentences, only 73% accuracy in sentence simplification is achieved as presented in Table 5.1. After a thorough analysis of each error

returned by the sentence simplification algorithm on the test dataset, most of the errors are due to failures in simplifying highly complex sentences. The sentence simplification algorithm has shortcomings in the handling of sentences that contain one or more clauses that share the subject or objects with the main clause, and/or contain other embedded clauses. This limitation often leads to incorrect and overspecified predicates and arguments, missed relation instances, and it also produces relation instances that are inconsistent with the information contained in the original sentence. For instance, from the sentence "በሻኪሶ ከተማ አካባቢ ነዋሪዎች በተነሳ ተቃውሞ ምክንያት ሥራውን እንዲያቋርጥ በተደረገው የሚድሮክ ወርቅ ኩባንያ ንብረት በሆነው የለገደንቢ ወርቅ ማምረቻ ላይ የካናዳ ከፍተኛ ባለሙያዎች ጥናት ሊያካሂዱ እንደሆነ ታወቀ፡፡" the relation generated by AOIE is: "ሥራውን እንዲያቋርጥ ጥናት ሊያካሂዱ እንደሆነ ታወቀ (Arg1:" በሻኪሶ ከተማ አካባቢ ነዋሪዎች", Arg2: "በተነሳ ተቃውሞ ምክንያት ")". The first argument of this extraction is incorrect, it should be "የካናዳ ከፍተኛ ባለሙያዎች". It also contains an overspecified predicate. There is also one missed relation (i.e., ነው (Arg1: "የለገደንቢ ወርቅ ማምረቻ", Arg2: "የሚድሮክ ወርቅ ኩባንያ", Arg3: "ንብረት)").

32.7% of the errors are due to errors in morphological analysis and POS tagging. For instance, from the sentence "በዓድዋ ጦርነት የተሰው ጀግኖችን አፅም በክብር ለማሳረፍ ታሰበ:", two relation instances were expected from the AOIE (i.e., ታሰበ (Arg1: "ጀግኖችን", Arg2: "አፅም", Arg3: "በክብር ለማሳረፍ"), ተሰው ("ጀግኖች", "በዓድዋ ጦርነት")"). However, since the morphological analyzer did not label the "የተሰው" as a relative verb, the relative clause is not detected and the sentence couldn't be simplified. As a result of this, only one instance of relation which contains an overspecified argument is extracted by AOIE system (i.e., "ታሰበ (Arg1:" በዓድዋ ጦርነት የተሰው ጀግኖችን ", Arg2: "አፅም", Arg3: "በክብር ለማሳረፍ ").

The remaining type of errors made by AOIE is due to erroneously chunked phrases. Incorrectly chunking the sentence often leads to incorrect predicate or arguments. For instance, the sentence: "የደደቢት እግር ኳስ ቡድን በኮሎኔል አወል አብዱራሂም ተመሰረተ" is chunked erroneously as: ([የደደቢት እግር ኳስ ቡድን] [በኮሎኔል አወል] [አብዱራሂም] [ተመሰረተ]) Thus, incorrect relation ተመሰረተ (Arg1:" የደደቢት እግር ኳስ ቡድን ", Arg2: -"በኮሎኔል አወል ", Arg3: -"አብዱራሂም ") is extracted instead of ተመሰረተ (Arg1:" የደደቢት እግር ኳስ ቡድን", Arg2: -"በኮሎኔል አወል አብዱራሂም ")".

CHAPTER SIX: CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, we have proposed an appropriate approach and architecture for AOIE system. To accomplish this, we reviewed existing approaches for OIE and analyzed their performance and applicability to Amharic text. In our comparison of approaches, we identified that the rule-based approach operating on deep parsed sentences yields the most promising results for OIE systems, as they enable higher precision. However, since there is no fully implemented dependency parser available for Amharic language, we have proposed a rule-based AOIE system that operates on shallow parsed sentences.

In order to evaluate the performance of the AOIE system, the proposed method is implemented in prototype software. We also have created a small 215 sentence corpus and hand-labeled the corpus for syntactic structure and extraction information. We have implemented an evaluator which automatically compares results of AOIE system to the manually annotated dataset and calculates precision. The result of the evaluation shows that AOIE achived promising result, 88% precision. From the result, we conclude that the complexity of sentences and the accuracy of parsing tools have a significant impact on the overall performance of AOIE.

6.2 Contribution

This work has several contributions to Amharic natural language processing.

We have contributed:

- A design for Amharic OIE.
- A chunking algorithm that utilizes POS and morphological information of words of a sentence.
- Amharic sentence simplification algorithm that breaks down long and complex Amharic sentences into short and simple sentences.
- Amharic relation extraction algorithm.

- Annotated dataset of 215 Amharic sentences extracted randomly from the online news sources.

6.3 Future Work

To further improve the performance of the AOIE system, we have identified the following directions for future work.

- Improvements to AOIE's text preprocessing have the potential to improve the system's usability. This includes the ability to resolve co-reference and recognize named entities.
- The performance of the AOIE system on highly complex sentences can significantly be improved by using deep parsing instead of shallow parsing.
- The sentence simplification algorithm can be improved by adding additional rules to handle highly complex sentences such as nested compound sentences.
- Noun-mediated relation extraction can be improved by identifying common nouns found between two nouns.
- Classifying the result returned by the OIE as concrete or abstract fact will make the integration of the OIE system with other downstream applications easier.
- Nesting the extracted relations allows to more accurately reflect the meaning of the original sentence.

REFERENCES

[1] F. Hogenboom, F. Frasincar, U. Kaymak, F. D. Jong, "An Overview of Event Extraction from Text," in *Proceedings of Detection, Representation and Exploitation of Events in the Semantic Web*, pp. 48-57, Bonn, Germany, 2011.

[2] S. Patwardhan, "Widening the field of view of information extraction through sentential event recognition", Unpublished PhD dissertation, School of Computing, University of Utah, 2010.

[3] R. Baradaran, B. Minaei-Bidgoli, "Event Extraction from Classical Arabic Text", *the International Arab Journal of Information Technology*, Vol. 12, No. 5, 2015.

[4] R. Grishman, B. Sundheim, "Message Understanding Conference – 6: A Brief History", In *Proceedings of the 16 International Conference on Computational Linguistics*, Copenhagen, Denmark, 1996.

[5] M. Banko, M.J. Cafarella, S. Soderland, M. Broadhead, O. Etzioni, "Open Information Extraction from the Web". In *Proceedings of the 20th international joint conference on Artificial intelligence(*IJCAI-07*),*pp. 2670-2676, Hyderabad, India, January 2007.

[6] A. Fader, S. Soderland, O. Etzioni, "Identifying relations for Open Information Extraction", in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011*), pp. 1535–1545, Edinburgh, UK, July 2011.

[7] S. Nam, Y. Hahm, S. Nam, K. Choi, "SRDF: Korean open information extraction using singleton property", In *Proceedings of International Semantic Web Conference, ISWC, Posters & Demonstrations Track*, 2015.

[8] M. Mausam, "Open information extraction systems and downstream applications", in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, pp. 4074-4077, New York, USA, July 2016.

[9] S. C. Abreu, T. L. Bonamigo, R. Vieira, "A review on relation extraction with an eye on Portuguese", *Journal of the Brazilian Computer Society*, 2013**.**

[10] A. Romadhony, D. H. Widyantoro, A. Purwarianti, "Using Relation Similarity on Open Information Extraction-Based Event Template Extraction", In *Proceedings of the 8th International Conference on Advanced Computer Science and Information Systems (*ICACSIS*)*, page.341-346 Malang, Indonesia, October 2016.

[11] N. Chambers, D. Jurafsky, "Template-based information extraction without the templates," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pp. 976–9868, Portland, Oregon, June 2011.

[12] N. Balasubramanian, S. Soderland, M. Mausam, O. Etzioni, "Generating coherent event schemas at scale". In *Proceedings of 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013),* Seattle, USA, October2013.

[13] O. Etzioni, A. Fader, J. Christensen, S. Soderland, M. Mausam, "Open Information Extraction: The Second Generation", in *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence (IJCAI -2011),* Barcelona, Catalonia, Spain, July 2011.

[14] R. Schneider, T. Oberhauser, T. Klatt, F. A. Gers, A. Lőser, "Analysing errors of open information extraction systems". In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems (BLGNLP-2017),* pp. 11-18, Copenhagen, Denmark, September 2017.

[15] J. Christensen, M. Mausam, S. Soderland, O. Etzioni, "Towards coherent multi-document summarization". In *proceedings of 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2013),* pp. 1163-1173, Atlanta, USA, 2013.

[16] J. Christensen, S. Soderland, G. Bansal, M. Mausam, "Hierarchical summarization: Scaling up multi-document summarization", In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 902–912, Maryland, USA, June 2014.

[17] D. Pighin, M. Cornolti, E. Alfonseca, K. Filippova, "Modeling events through memory-based, open-IE patterns for abstractive summarization", In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 892–901, Maryland, USA, June 2014.

[18] G. Stanovsky, I. Dagan, M. Mausam, "Open IE as an intermediate structure for semantic tasks", In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 303–308, Beijing, China, July 2015.

[19] S. Soderland, J. Gilmer, R. Bart, O. Etzioni, D. S. Weld, "Open information extraction to KBP relations in 3 hours", In *Proceedings of the Sixth Text Analysis Conference (*TAC *2013),* Maryland, USA, November 2013.

[20] A. Fader, L. Zettlemoyer, O. Etzioni, "Open Question Answering over Curated and Extracted Knowledge Bases", In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1156–1165, New York, USA, August 2014.

[21] W. E. Zhang, E. Abebe, Q. Z. Sheng, K. Taylor, "Towards building open knowledge base from programming question-answering communities", In *Proceedings of the 15th International Semantic Web Conference (ISWC 2016),* Kobe, Japan, October 2016.

[22] T. Khot, A. Sabharwal, P. Clark, "Answering complex questions using open information extraction", In *Proceedings* of the *55th Annual Meeting* of the *Association* for *Computational Linguistics*, pp. 311–316. Vancouver, Canada, July 2017.

[23] A. Ritter, M. Mausam, O. Etzioni, "A Latent Dirichlet Allocation Method for Selectional Preferences", In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,* pp. 424–434, Uppsala, Sweden, July 2010.

[24] T. Lin, M. Mausam, O. Etzioni, "Identifying Functional Relations in Web Text", In *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* EMNLP'10, Stroudsburg, USA, 2010.

[25] S. Schoenmackers, O. Etzioni, D. S. Weld, J. Davis, "Learning first-order horn clauses from web text", In *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* EMNLP'10, pp. 1088–1098, Stroudsburg, USA, 2010.

[26] J. Berant, I. Dagan, J. Goldberger, "Global learning of typed entailment rules", In *Proceedings of the 49th* Annual Meeting of *the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, June 2011.

[27] B. Gambäck, "Tagging and Verifying an Amharic News Corpus", In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, Turkey, May 2012.

[28] B. Gamback, L. Asker, "Experiences with Developing Language Processing Tools and Corpora for Amharic", In *Proceedings of the 5th conference on regional: impact of information society technologies in Africa*, Durban, South Africa, May 2010.

[29] F. Wu, D. S. Weld, "Open Information Extraction using Wikipedia", In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics,* pp. 118-127, Uppsala, Sweden, July 2010.

[30] M. Mausam, M. Schmitz, R. Bart, S. Soderland, O. Etzioni, "Open Language Learning for Information Extraction", In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, Association for Computational Linguistics*, pp 523–534, Jeju Island, Korea, 2012.

[31] Y. H. Tseng, L. H. Lee, B. S. Liao "Chinese Open Relation Extraction for Knowledge Acquisition", in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, EACL (2014), pp 12–16, Gothenburg, Sweden, April 2014.

[32] Getasew Tsedalu, "Information Extraction model from Amharic News texts", Unpublished Masters Thesis, Department of Computer Science, Addis Ababa University, 2010.

[33] Bekele Worku, "Information Extraction from Amharic language Text: Knowledge-poor approach", Unpublished Masters Thesis, Department of Computer Science, Addis Ababa University, 2015.

[34] Baye Yimam, የአማርኛ ሰዋስው (Amharic Grammar). Addis Ababa, Ethiopia,2000.

[35] M. Gasser, "A dependency grammar for. Amharic*"*. In *Proceedings of the Workshop* on *Language Resources and Human Language*. *Technologies for Semitic Languages*, 2010

[36] G. Amare:" ዘመናዊ የአማርኛ ሰዋስው በቀላል አቀራረብ (Modern Amharic Grammar in a Simple Approach)", Addis Ababa, Ethiopia, 2010.

[37] N. Konstantinova, "Review of Relation Extraction Methods: What Is New Out There?", In *Proceedings of International data science conference AIST'14*, pp. 15–28, 2014.

[38] *S. Sarawagi*, "Information Extraction*", Foundations and Trends in Databases*, v.1 n.3, pp. 261-377, 2008.

[39] J. Piskorski, R. Yangarber, "Information Extraction: Past, Present and Future", Multi-source, Multilingual Information Extraction and Summarization 11, Theory and Applications of Natural Language Processing, Germen, Berlin, 2013.

[40] P. Andersen, P. Hayes, A. Huettner, L. Schmandt, I. Nirenburg, S. Weinstein, "Automatic extraction of facts from press releases to generate news stories", In *Proceedings of the 3rd Conference on Applied Natural Language Processing, ANLC '92*, pp. 170–177, USA, Stroudsburg, 1992.

[41] N. T. Nakashole, "Automatic Extraction of Facts, Relations, and Entities for Web-Scale Knowledge Base Population", Unpublished PHD dissertation, Faculty of Natural Sciences and Technology, University of saarland max planck institute for informatics, Germany, 2012.

[42] K. Kaiser, S. Miksch, "Information Extraction. A Survey", Report for Asgaard-TR-2005-6, 2005.

[43] Y. Shinyama and S. Sekine, "Preemptive information extraction using unrestricted relation Discovery", *In Proceedings of Human Language Technology Conference and North American chapter of the Association for Computational Linguistics (HLT-NAACL 2006)*, USA, New York,2006.

[44] M. Banko, "Open Information Extraction for the Web", Unpublished PHD dissertation, Department of Computer Science, University of Washington, 2009.

[45] A. Bassa, "GerIE: Open Information Extraction for German Texts", Unpublished Master's Thesis, Knowledge Technologies Institute, Graz University of Technology, 2016.

[46] D. T. Vo and E. Bagheri, "Open Information Extraction", ENCYCLOPEDIA WITH SEMANTIC COMPUTING, Vol. 1, No. 1, 2016.

[47] M. Banko, O. Etzioni, "The tradeoffs between open and traditional relation extraction", in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pages 28-26, Columbus, Ohio, USA, June 2008.

[48] J. J. M. Raposo, "Open Information Extraction from Dialogue Transcriptions", Unpublished Master's Thesis, Information Systems and Computer Engineering, 2016.

[49] HaBiT - Harvesting big text data for under-resourced languages, `http://www.habit-project.eu/`

[50] P. Rychlý, V. Suchomel, "Annotated Amharic Corpora", in *International Conference on Text, Speech, and Dialogues*, pages 295-302. Springer, Cham, 2016.

[51] Abeba Ibrahim, Yaregal Assabie, "Amharic Sentence Parsing Using Base Phrase Chunking", In *International Conference on Intelligent Text Processing and Computational Linguistics,* pages 297-306. Springer, Berlin, Heidelberg, 2014.

[52] Binyam Ephrem, Yusuke Miyao, Baye Yimam, "Universal Dependencies for Amharic". In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, pages 2216– 2222. Miyazaki, Japan: European Language Resources Association (ELRA), 2018.

[53] M. Gasser, "HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya", *In Conference on Human Language Technology for Development*, pages 94-99, Alexandria, Egypt, 2011.

[54] Poornima C, Dhanalakshmi V, "Rule based Sentence Simplification for English to Tamil Machine Translation System", *International Journal of Computer Applications* (0975 – 8887) Volume 25– No.8, July 2011.

[55] C. Niklaus, B. Bermeitinger, S. Handschuh, A. Freitas, "A sentence simplification system for improving relation extraction", In *Proceedings of COLING 2016: System Demonstrations, The 26th International Conference on Computational Linguistics*, Osaka, Japan, pages 170–174, December 11-16, 2016.

[56] C. C. Xavier, M. Souza, "Open Information Extraction Based on Lexical-Syntactic Patterns", In *Proceedings of the 2013 Brazilian Conference on Intelligent Systems*, Fortaleza, Brazil, October 2013.

[57] P. Gamallo, M. Garcia, S. Fernández-Lanza, "Dependency-based open information extraction", In *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*, pages 10-18, Avignon, France, April 2012.

[58] Luciano Del Corro, R. Gemulla. "ClausIE: Clause-based Open Information Extraction". In *Proceedings of the 22nd International Conference on World Wide Web, WWW'13. Association for Computating Machinery*, 2013.

[59] D. S. Batista, "Large-Scale Semantic Relationship Extraction for Information Discovery", Unpublished PHD dissertation, Technical Superior Institute, University of Lisbon, 2015.

[60] M. Grap, "A hybrid approach to general information extraction", Unpublished Masters Thesis, Department of Computer Science, California Polytechnic State University, 2015.

[61] C. Cortes, V. Vapnik, "Support vector networks", In *Proceedings of Machine*. Learning, vol. 20, pp. 273-297, 1995.

[62] J. D. Lafferty, A, McCallum, F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", *In Proceedings of the Eighteenth International Conference on Machine Learning, ICML'01. Association for Computing Machinery*, 2001.

[63] *M Mintz*, *S Bills*, *R Snow*, *D Jurafsky*, "Distant supervision for relation extraction without labeled data", in *Proceedings* of the *Joint Conference* of the *47th Annual Meeting of the ACL and the 4th International* Joint Conference *on Natural Language Processing of the AFNLP,* 2009.

[64] M. J. Cafarela, D. Downey, S. Soderland, O. Etzioni, "KnowItAll: Fast, scalable information extraction from the Web.", *In Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp. 563-570, 2005.

[65] M. Yahya, S. Whang, R. Gupta, A. Y. Halevy, "Renoun: Fact extraction for nominal attributes", In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014),* pp, 325–335, Doha, Qatar, October 2014.

[66] D. Truong, D. Vo, and U.T Nguyen, "Vietnamese Open Information Extraction", In SoICT '17, NY, USA, December 2017.

[67] The Reporter Ethiopia, `https://www.ethiopianreporter.com`

[68] Walta Media and Communication Corporate, `http://www.waltainfo.com`

APPENDICES

Appendix A: The Amharic Alphabet

Order							Labialized				
1st	2nd	3rd	4th	5th	6th	7th					
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ					
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ			ሏ		
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ					
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ			ሟ		
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ					
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ			ሯ		
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ			ሷ		
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ			ሿ		
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ	ቊ	ቋ	ቌ	ቍ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ			ቧ		
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ			ቷ		
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ			ቿ		
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኈ	ኊ	ኋ	ኌ	ኍ
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ			ኗ		
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ			ኟ		
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ					
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ					
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ					
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ	ኰ	ኲ	ኳ	ኴ	ኵ
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ					
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ			ዟ		
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ			ዧ		
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ					
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ			ዷ		
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ					
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ	ጒ	ጓ	ጔ	ጕ
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ			ጧ		
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ			ጯ		
ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ			ጷ		
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ					
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ			ፏ		
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ					

ᎀ	ᎁ	ᎂ	ᎃ	ᎄ	ᎅ	ᎆ

Appendix B: Sample POS and morphological tagged, and chunked sentences

1. [ፍርድ N ቤቶች N] [የበጀትና NP የግዥ NP ችግር N] [እንዳለባቸው PRV ተገለጸ PAV]
2. [ፕሬዚዳንቱ ND] [አደጋው ND] [የኢትዮጵያን NP ህዝብ N] [እንዳሳዘነ PRV ገለጹ PV]
3. [አመልድ N] [ለ300ሺ NP ወገኖች N ኢርዳታ N] [አከፋፈለ V]
4. [ቦረና N ውስጥ POSTP] [የኦህዴድ NP ተሃድሶ N ውይይት N] [ተጀመረ PAV]
5. [አለምአቀፍ ADJ የሩጫ NP ውድድር N] [በአዲስአበባ NP] [ሊካሄድ PAV ነው V]
6. [የጋምቤላ NP ፖርክን ND ለማሻሻል INF] [110ሺ ADJ ብር N] [ተመደበ PAV]
7. [ከኤርትራ NP 205 ADJ ኢትዮጵያውያን N] [ወደ PREP አገራቸው ND] [ተመለሱ PAV]
8. [በእንግሊዝ NP የሚኖሩ IRV ኢትዮጵያውያን N] [የጋራ NP መድረክ N] [መሰረቱ V]
9. [ዲስትሪክቱ ND] [በ36ሚሊየን NP ብር N] [የመንገድ NP ጥገና N] [አከናውኗል GAV]
10. [የኢትዮጵያ NP የትምህርት NP ሽፋን N] [በ6ነጥብ4በመቶ NP] [አደገ PV]
11. [ሕብረት NP ኢንሹራንስ ND] [ካፒታሉን ND] [ወደ PREP ግማሽ ADJ ቢሊዮን ADJ ብር N] [አሳደገ PV]
12. [ኢማተልፈ N ያስገነባቸው IRV 17 ADJ ትምህርትቤቶች N] [አገልግሎቶችን ND መስጠት INF] [ጀመሩ PV]
13. [አረጋውያን ND] [ራሳቸውን ND የሚያቋቁሙበትን IRV ማህበር N] [መሰረቱ V]
14. [ሰራተኞቹ ND][ድርጅቱ ND የሸለመውን PRV ኮከብ ADJ ሰራተኛ N][ሊቀመንበር N][አደረጉት PV]
15. [ለ PREP 29 ADJ ኛው N] [የመሬት NP ሊዝ N ጨረታ N] [አዲስ ADJ ሶፍትዌር N] [እየተዘጋጀ PPRV ነው V]
16. [ብሔራዊ ADJ አልኮል N ፋብሪካ N] [ወደ PREP ግል ADJ ይዞታ N ለማዛወር INF] [ጨረታ N] [ወጣ PV]
17. [ካስትል NP ግሩፕ N] [የዘቢዳር NP ቢራን ND] [58 ADJ በመቶ NP ድርሻ N መግዛቱን INF] [ይፋ N] [አደረገ V
18. [አየር N መንገድ N] [አዲስ ADJ የተንቀሳቃሽ NP ስልክ N መተግበሪያ N] [ይፋ N] [አደረገ PV]
19. [ሱዳን ND] [ወደ PREP ኢትዮጵያ N] [የባቡር NP መስመር N] [እንደምትዘረጋ IRPV አስታወቀች PV]
20. [የኢትዮጵያ NP አየር N መንገድ N በረራዎች N] [ለሁለት NP ሰዓታት N] [ተቋረጡ PAV]
21. [በዓድዋ NP ጦርነት N የተሰው V ጀግኖችን ND] [አፅም N] [በክብር NP ለማሳረፍ INF] [ታሰበ PAV]
22. [በፋይናንስ NP አቅርቦት N ችግር N] [የ PREP 15 ADJ ባለ N ኮከብ N ሆቴሎች N ግንባታ N] [እየተጓተተ PPRV ነው V]
23. [ኬኬ ND] [በጣና NP ሐይቅ N ላይ POSTP የተንሰራፋውን V] [እምቦጭ NP አረም N ለማስወገድ INF] [20 ADJ ሚሊዮን ADJ ብር N] [ዕርዳታ NP] [ሰጠ PV]
24. [በኮሎኔል NP አወል N አብዱራሂም N የተመሠረተው PPRV የደደቢት NP እግር N ኳስ N ቡድን N] [ወደ PREP ፕሪሚየር N ሊግ N ካደገ PRV በኋላ POSTP] [የክለቦች NP ጥሎ N ማለፍ INF] [ዋንጫ N ማንሳት INF] [ችሎ GV ነበር V]
25. [አልማዝ ND] [ከሃረር NP እየመጣች PRV ነው V] [ግን N] [አበበ ND] [እቤት NP ውስጥ POSTP] [የለም V]
26. [አበበ N] [ወደ PREP ሃረር ND ሊሄድ IV ይችላል AV] [ወይም CONJ] [አልማዝ N] [ከሃረር NP] [ልትመጣ IV ትችላለች V]
27. [የተረጎምኩት NP መፅሃፍ N ተመሳስሎ PAV ተተርጉሞበኛል PGAV ያሉት PRV ደራሲ N] [194 ADJ ሺ ADJ ብር N ካሣ N] [ተፈረደላቸው PAV]
28. [የጠዋት NP ጸሃይ N ስትወጣ IV] [የእግር NP ጉዞ N ማድረግ INF እወዳለው V] [ነገር N ግን CONJ] [ዛሬ ADV] [ዝናብ N ስለነበረ PRV እስከ PREP ምሳ N] [ሰሃት NP ድረስ POSTP] [ከቤት NP አልወጣሁም PV]

29. [ወደ PREP ውጭ N የሚላከው IRPV የማር NP ምርት N] [በከፍተኛ NP ሁኔታ N እያሽቆለቆለ V መምጣቱ INF] [ተገለጸ PAV]
30. [በኢትዮጵያውያን NP የተቋቋመው PPRV የሞባይልና NP የኮምፒዩተር NP መገጣጠሚያ N] [በ PREP 50 ADJ ሚሊዮን ADJ ብር N ኢንቨስትመንት N] [ሥራ N] [ጀመረ PV]
31. [ብርሃን N ባንክ N] [መጠነኛ ADJ የትርፍ NP ቅናሽ N ባስመዘገበበት PRV ዓመት N የተስፋፋ PPRV የባንክ NP እንቅስቃሴ N] [እንደነበረው PAV አስታወቀ PV]
32. [ምክትል ADJ ከንቲባው ND] [የራይድ NP ትራንስፖርት N አገልግሎት N እንዲቀጥል IV የመፍትሔ V አማራጭ N] [አስቀመጡ PV]
33. [በተጠናቀቀው PPRV የጥቅምት NP ወር N] [በኢትዮጵያ NP የምርት NP ገበያ N] [የቡና NP ግብይት N መጠን N] [ከቀዳሚው NP ወር N] [በ PREP 74 ADJ በመቶ NP ጭማሪ ADJ] [የታየበት PPRV ሆኗል GAV]
34. [ለዓመታት NP በውጣ NP ውረዶች N የተፈተነው PPRV የኢትዮጵያ NP ትራንስፖርት N አሠሪዎች AN ፌዴሬሽን ND] [ምሥረታ N] [ተከናወነ PAV]
35. [ከ PREP 1.2 ADJ ቢሊዮን ADJ ዶላር N በላይ POSTP] [ገቢ AN የታቀደለት PPRV ቡና N] [በሦስት NP ወራት N] [204 ADJ ሚሊዮን ADJ ዶላር N] [አስገኝቷል GAV]
36. [ኤርትራ ND] [ፕሬዚዳንት N የሆነችበት PRV የአፍሪካ NP ቀንድ N ሠራተኞች N ማኅበራት N ኮንፌዴሬሽን N ምሥረታ N] [በአዲስ NP አበባ N ዕውን N] [ተደረገ PAV]
37. [የሕብረት ባንክ NP የአክሲዮን NP ባለቤቶች N] [የባንኩ NP ጠቅላላ ADJ ካፒታል N] [አምስት ADJ ቢሊዮን ADJ ብር N] [እንዲሆን IV ወሰኑ PV]
38. [የቀላል V ባቡር N ፕሮጀክት N ለመጠገን INF] [70 ADJ ሚሊዮን ADJ ብር N] [ይጠይቃል AV ተባለ V]
39. [በሐሰተኛ NP ደረሰኝ N] [ከአራት NP ቢሊዮን ADJ ብር N በላይ POSTP] [ግብይት N ፈጽመዋል GAV የተባሉ V] [124 ADJ ድርጅቶች N] [እየተመረመሩ PPRV ነው V]
40. [የቱርክ ኩባንያ][በትግራይ ክልል][በ 750 ሚሊዮን ዩሮ ኢንዱስትሪ ፓርክ ለመገንባት][የኢንቨስትመንት ፈቃድ ማውጣት][ጀመረ]
41. [ሐሰተኛ ADJ ደረሰኝ N አትመው GV በመሸጥ INF] [ከአራት NP ቢሊዮን ADJ ብር N በላይ POSTP] [ግብይት N የፈጸሙ PRV 124 ADJ ድርጅቶች N] [ይፋ ADJ ተደረጉ N]
42. [ዓባይ AN ባንክ N] [ከ PREP 400 ADJ ሚሊዮን ADJ ብር N በላይ POSTP] [ትርፍ N ቢያስመዘግብም IV ዓለም N አቀፍ N ባንኮች N የወጣባቸው PRV ሕግ N] [ጫና ADJ] [እንዳሳደረበት PRV ገለጸ PV]

Appendix E: Automatically Extracted Relations

1. አደረገ (Arg1:" አየር መንገድ“, Arg2:-"አዲስ የተንቀሳቃሽ ስልክ መተግበሪያ ",Arg3:-"ይፋ ")
2. እንደምትዘረጋ አስታወቀች (Arg1:" ሱዳን ", Arg2:-"ወደ ኢትዮጵያ ",Arg3:-"የባቡር መስመር ")
3. ተቋረጡ (Arg1:" የኢትዮጵያ አየር መንገድ በረራዎች", Arg2:-"ለሁለት ሰዓታት ")
4. እንዲሆን ወሰኑ (Arg1:" የሕብረት ባንክ የአክሲዮን ባለቤቶች", Arg2:-"የባንኩ ጠቅላላ ካፒታል ",Arg3:-"አምስት ቢሊዮን ብር ")
5. ታሰበ (Arg1:" በዓድዋ ጦርነት የተሰው ጀግኖችን", Arg2:-"አፅም ",Arg3:-"በክብር ለማሳረፍ ")
6. እንዳለባቸው ተገለጸ (Arg1:" ፍርድ ቤቶች " ,Arg2:-"የበጀትና የግዥ ችግር ")
7. እየተጓተተ ነው (Arg1:" በፋይናንስ አቅርቦት ችግር " ,Arg2:-"የ 15 ባለ ኮከብ ሆቴሎች ግንባታ ")
8. እየተዘጋጀ ነው (Arg1:" ለ 29 ኛው " ,Arg2:-"የመሬት ሊዝ ጨረታ ",Arg3:-"አዲስ ሶፍትዌር ")
9. ወጣ (Arg1:" ብሔራዊ አልኮል ፋብሪካ " ,Arg2:-"ወደ ግል ይዞታ ለማዛወር ",Arg3:-"ጨረታ ")
10. አደረገ (Arg1:" ካስትል ግሩፕ " ,Arg2:-"የዘቢዳር ቢራን ",Arg3:-"58 በመቶ ድርሻ መግዛቱን ",Arg4:-"ይፋ ")
11. ሰጠ (Arg1:" ኬኬ " ,Arg2:-"በጣና ሐይቅ ላይ የተንሰራፋውን ",Arg3:-"እምቦጭ አረም ለማስወገድ ",Arg4:-"20 ሚሊዮን ብር ",Arg5:-"ዕርዳታ ")
12. እንደነበረው አስታወቀ (Arg1:" ብርሃን ባንክ ዓመት " ,Arg2:-"የባንክ እንቅስቃሴ ")
13. አስቀመጡ (Arg1:" ምክትል ከንቲባው " ,Arg2:-"የራይድ ትራንስፖርት አገልግሎት እንዲቀጥል የመፍትሔ አማራጭ ")
14. የታየበት ሆኗል (Arg1:" ወር " ,Arg2:-"በኢትዮጵያ የምርት ገበያ ",Arg3:-"የቡና ግብይት መጠን ",Arg4:-"ከቀዳሚው ወር ",Arg5:-"በ 74 በመቶ ጭማሪ ")
15. ተከናወነ (Arg1:" የኢትዮጵያ ትራንስፖርት አሠሪዎች " ,Arg2:-"ፌዴሬሽን ",Arg3:-"ምሥረታ ")
16. እንዳሳደረበት ገለጸ (Arg1:" ዓባይ ባንክ " ,Arg2:-"ከ 400 ሚሊዮን ብር በላይ ",Arg3:-"ሕግ ",Arg4:-"ጫና ")
17. ይጠይቃል ተባለ (Arg1:" የቀላል ባቡር ፕሮጀክት ለመጠገን " ,Arg2:-"70 ሚሊዮን ብር ")
18. እየተመረመሩ ነው (Arg1:" በሐሰተኛ ደረሰኝ " ,Arg2:-"ከአራት ቢሊዮን ብር በላይ ",Arg3:-"124 ድርጅቶች ")
19. ይፋ ተደረጉ (Arg1:" ሐሰተኛ ደረሰኝ አትመው በመሸጥ " ,Arg2:-"ከአራት ቢሊዮን ብር በላይ ",Arg3:-"124 ድርጅቶች ")
20. ጀመረ (Arg1:" የቱርክ ኩባንያ " ,Arg2:-"በትግራይ ክልል ",Arg3:-"በ 750 ሚሊዮን ዩሮ ኢንዱስትሪ ፓርክ ለመገንባት ",Arg4:-"የኢንቨስትመንት ፈቃድ ማውጣት ")
21. ተደረገ (Arg1:" ኤርትራ " ,Arg2:-"የአፍሪካ ቀንድ ሠራተኞች ማኅበራት ",Arg3:-"ኮንፌዴሬሽ ",Arg4:-"ምሥረታ ",Arg5:-"በአዲስ አበባ ዕውን ")
22. አስገኝቷል (Arg1:" ከ 1.2 ቢሊዮን ዶላር በላይ " ,Arg2:-"ቡና ",Arg3:-"በሦስት ወራት ",Arg4:-"204 ሚሊዮን ዶላር ")
23. ጀመረ (Arg1:" የሞባይልና የኮምፒዩተር መገጣጠሚያ " ,Arg2:-"በ 50 ሚሊዮን ብር ኢንቨስትመንት ሥራ ")
24. ችሎ ነበር (Arg1:" የደደቢት እግር ኳስ ቡድን በኋላ ",Arg2:-"የክለቦች ጥሎ ማለፍ ",Arg4:-"ዋንጫ ማንሳት ")
25. ተገለጸ (Arg1:" የማር ምርት " ,Arg2:-"በከፍተኛ ሁኔታ እያሽቆለቆለ መምጣቱ ")
26. እንዳሳዘነ ገለጹ (Arg1:" ፕሬዚዳንቱ " ,Arg2:-"አደጋው ",Arg3:-"የኢትዮጵያን ህዝብ ")
27. አከፋፈለ (Arg1:" አመልድ " ,Arg2:-"ለ300ሺ ወገኖች ኢርዳታ ")
28. ተጀመረ (Arg1:" ቦረና ውስጥ " ,Arg2:-"የኦህዴድ ተሃድሶ ውይይት ")
29. ሊካሄድ ነው (Arg1:" አለምአቀፍ የሩጫ ውድድር " ,Arg2:-"በአዲስአበባ ")
30. ተመደበ (Arg1:" የጋምቤላ ፖርክን ለማሻሻል " ,Arg2:-"110ሺ ብር ")
31. ተመለሱ (Arg1:" ከኤርትራ 205 ኢትዮጵያውያን " ,Arg2:-"ወደ አገራቸው ")
32. ጀመሩ (Arg1:" ኢማተልፈ ያስገነባቸው ", Arg2:-"17 ትምህርትቤቶች ",Arg3:-"አገልግሎቶችን መስጠት ")
33. መሰረቱ (Arg1:" አረጋውያን ", Arg2:-"ማህበር ")
34. አደረጉት (Arg1:" ሰራተኞቹ ", Arg2:-"ኮከብ ሰራተኛ ",Arg3:-"ሊቀመንበር ")
35. ተፈረደላቸው (Arg1:" ደራሲ", Arg2:-"194 ሺ ብር ካሣ ")

36. አሳደገ (Arg1:" ሕብረት ኢንሹራንስ " ,Arg2:-"ካፒታሉን ",Arg3:-"ወደ ግማሽ ቢሊዮን ብር ")
37. መሰረቱ (Arg1:" ኢትዮጵያውያን " ,Arg2:-"የጋራ መድረክ ")
38. አከናውኗል (Arg1:" ዲስትሪክቱ " ,Arg2:-"በ36ሚሊየን ብር ",Arg3:-"የመንገድ ጥገና ")
39. አደገ (Arg1:" የኢትዮጵያ የትምህርት ሽፋን " ,Arg2:-"በ6ነጥብ4በመቶ ")
40. ተስፋፋ (Arg1:" ዓመት " ,Arg2:-"የባንክ እንቅስቃሴ ",Arg3:-"መጠነኛ የትርፍ ቅናሽ ")
41. ተፈተነ (Arg1:" የኢትዮጵያ ትራንስፖርት አሠሪዎች " ,Arg2:-"ለዓመታት በውጣ ውረዶች ")
42. ወጣ (Arg1:" ትርፍ ቢያስመዘግብም ዓለም አቀፍ ባንኮች " ,Arg2:-"ሕግ ")
43. ፈጽመዋል ተባለ (Arg1:" 124 ድርጅቶች ግብይት ")
44. ፈጸመ (Arg1:" ግብይት " ,Arg2:-"124 ድርጅቶች ")
45. ሆነ (Arg1:" ፕሬዚዳንት " ,Arg2:-"የአፍሪካ ቀንድ ሠራተኞች ",Arg3:-"ማኅበራት ")
46. ታቀደ (Arg1:" ቡና ገቢ ")
47. ተቋቋመ (Arg1:" የሞባይልና የኮምፒዩተር መገጣጠሚያ " ,Arg2:-"በኢትዮጵያውያን ")
48. ተመሰረተ (Arg1:" የደደቢት እግር ኳስ ቡድን " ,Arg2:-"በኮሎኔል አወል ",Arg3:-"አብዱራሂም ")
49. የለም (Arg1:" አበበ " ,Arg2:-"እቤት ውስጥ ")
50. ልትመጣ ትችላልች (Arg1:" አልማዝ " ,Arg2:-"ከሃረር ")
51. ሸለመ (Arg1:" ድርጅቱ " ,Arg2:-"ኮከብ ሰራተኛ ")
52. አለ (Arg1:" የተረጎምኩት መፅሃፍ ተመሳስሎ ተተርጉሞብኛል ደራሲ "))
53. HAS ("ባቡር"," መስመር")
54. HAS ("ኢትዮጵያ"," አየር መንገድ በረራዎች")
55. HAS ("አክሲዮን"," ባለቤቶች")
56. HAS ("ባንክ"," ጠቅላላ ካፒታል")
57. HAS ("መሬት"," ሊዝ ጨረታ")
58. HAS ("ዘቢዳር"," ቢራን")
59. HAS ("ራይድ"," ትራንስፖርት አገልግሎት")
60. HAS ("ምርት"," ገበያ")
61. HAS ("ቡና"," ግብይት መጠን")
62. HAS ("ቱርክ", " ኩባንያ")
63. HAS ("ኢንቨስትመንት", " ፈቃድ")
64. HAS ("ደደቢት", " እግር ኳስ")
65. HAS ("ኢትዮጵያ"," ህዝብ")
66. HAS ("ሩጫ"," ውድድር")
67. HAS ("ጋምቤላ"," ፖርክ")
68. HAS ("መንገድ"," ጥገና")
69. HAS ("ደደቢት"," እግር ኳስ ቡድን")
70. HAS ("ጠዋት"," ጸሃይ")

Declaration

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

Declared by:

Name: ___Seble Girma_____

Signature: _____

Date: _____March 16, 2020_____

Confirmed by advisor:

Name: ____Yaregal Assabie (PHD)_____

Signature: ____ _____

Date: ___March 16, 2020_____