ADDIS ABABA UNIVERISTY

COLLEGE OF NATURAL SCIENCES

Semantic Role Labeler for Amharic Text Using Memory Based Learning

Eskedar Yirga

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

June, 2017

ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCE

Semantic Role Labeler for Amharic Text Using Memory Based Learning

Eskedar Yirga

Advisor: Yaregal Assabie (PhD)

This is to certify that the thesis prepared by Eskedar Yirga, titled: *Semantic Role Labeler for Amharic Text Using Memory Based Learning* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	Name	**Signature**	**Date**
1.	Advisor: Yaregal Assabie(PhD),		
2.	Examiner: Mulugeta Libsie (PhD)		
3.	Examiner: Dida Midekso (PhD)		

ABSTRACT

Human knowledge is recoded in natural language. The records are kept in computers or on paper to be manipulated and reserved for use in the future. Then natural language processing plays an important role to accomplish computer domain-independent understanding of natural language. A useful step towards that role is assigning semantic roles to the constituents of a sentence, which refers to semantic role labeling. It allows one to recognize semantic arguments of a situation, even when expressed in different syntactic configurations. Still now there is no automatic semantic role labeler for Amharic adopting semantic role labeler of other languages for Amharic language is difficult since Amharic language is morphologically complex and the general role of language specific characteristics in the extraction of semantic content is different. Therefore, the objective of this study was to develop semantic role labeler for Amharic text using memory based learning algorithm. The system was trained on 551 instances. To evaluate the result leave one out cross-validation technique was employed. The evaluation result showed that the accuracy in classifying semantic role achieved 82.51% accuracy and 78.19% F-Score with default parameter and 89.29% accuracy and 79.77% F-Score with optimize parameter setting.

Keywords: Semantic role labeler, memory based learning, treebank, cross validation, feature extraction.

ACKNOWLEDGMENTS

My utmost gratitude goes to all those who have helped me to make the completion of this paper possible.

Firstly, I would like to thank the almighty God for giving me strength and determination to complete this thesis.

Secondly, I would like to express my deepest gratitude to my advisor Yaregal Assabie (PhD) for his generous and fruitful advice, fatherly encouragement, and willingness to help me all the time and for his critical evaluation of my thesis and giving constructive comments.

I would like to convey my great gratitude to Getahun Amare (PhD), Ato Binyam Eframe and Kibebe Tsehay (PhD) for going through the questions that concern about linguistic aspect in order for evaluating its content validity and giving their professional comment on it.

I am deeply indebted to the support of Ato Yitayal Abate, Ato Kidane Woldemariyam and my sister Tseganesh Yirga for giving me the courage and inspiration to undertake this endeavor from beginning to end.

Last but not least, I thank my family and friends for their patience and support through this journey.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Acronyms and Abbreviations

IB1	Instance Based one
IB2	Instance Based Two
ID	Inverse Distance
IE	Information Extraction
IG	Information Gain
IGTREE	Information Gain Tree
ILK	Linguistic Knowledge
IR	Information Retrieval
k-NN	k-nearest neighbor
MBL	Memory Based Learning
ML	Memory Learning
MVDM	Modified value difference metric
PARA	Predicate-Argument Recognition Algorithm
POS / PoS	Part of speech tagging
SRL	Semantic Role Labeler
SVMs	Support Vector Machines
TiMBL	Tilburg Memory Based Learner
TRIBL2	Tree Instance Based Learning Two
TTS	Text to speech
VP	Verb phrase
WSD	Word Sense Disambiguation

CHAPTER ONE: INTRODUCTION

1.1 Background

Semantics is a field of Natural Language Processing (NLP) concerned with extracting meaning from a sentence. Semantic role labeling, sometimes also referred to as shallow semantic parsing, takes the initial steps in extracting meaning from text by giving generic labels or roles to the words of the text, the meaning of this small set of labels can be assumed to be understood by the machine [1]. Semantic role labeling is a leading task of identifying arguments for a predicate and assigning semantically meaningful labels to them [2].

Understanding events and their participants is a key part of understanding natural language. At a high level, understanding an event means being able to answer the question "Who did what to whom" (and perhaps also "when and where"). The answers to this question may be expressed in many different ways in a sentence [3]. For example, if we want to process sentences to help us answer a question about a purchase of stock by XYZ corporation, we need to understand this event despite many different surface forms. The event could be described by a verb (sold, bought) or a noun (purchase) and XYZ corporation can be the syntactic subject (of bought), the indirect object (of sold), or in a genitive or noun compound relation (with the noun purchase), in the following sentences, despite having the same notational role in all of them:

- XYZ Corporation bought the stock.
- They sold the stock to XYZ Corporation.
- The stock was bought by XYZ Corporation.
- The purchase of the stock by XYZ Corporation.
- The stock purchase by XYZ Corporation.

Semantic role labeler (SRL) lets us capture the commonality between these sentences [3]. We will be able to represent the fact that there was a purchase event, that the participants in this event were XYZ corporation and some stock, and that XYZ corporation played a specific role, the role of acquiring the stock. This shallow semantic representation level is called semantic role [3]. Semantic roles are representations that express the abstract role that arguments of a predicate can

take in the event. These can be very specific like the BUYER, abstract like the AGENT, or super-abstract (the PROTO-AGENT). These roles can both represent general semantic properties of the arguments and also express their likely relationship to the syntactic role of the argument in the sentence. AGENTS tend to be the subject of an active sentence, THEMES the direct object, and so on; these relations are codified in databases like proposition Bank (PropBank) and FrameNet. Semantic role representations have many potential applications in natural language processing and have recently been shown to be beneficial in question answering [4, 5], textual entailment [6], machine translation [7, 8], dialogue systems [9] and event extraction [14] among others.

The main reason that computational systems use semantic roles is to act as a shallow meaning representation that can let us make simple inferences that aren't possible from the pure surface string of words, or even from the parse tree [7]. This shallow semantics might act as a useful intermediate language in machine translation. Many of the most glaring errors made by today's statistical machine translation systems are those resulting from confusion of semantic roles. Translation errors of this type frequently result in critical misunderstandings of the essential meaning of the original input language sentences who did what to whom, for whom or what, how, where, when, and why. Wu and Fung [7], for the first time demonstrate that shallow semantic parsing can improve translation accuracy of static machine translation models. Evidence has begun to accumulate that semantic frames predicates and semantic roles tend to preserve consistency across translations better than syntactic roles do. This is, of course, by design; it follows from the definition of semantic roles, which are less language dependent than syntactic roles [7].

Sematic role labeler (SRL) has obvious applications for template-filling tasks such as information extraction and question answering. The Shen and Lapata [4] assess the contribution of semantic role labeling to open-domain factoid question answering. The authors present a graph-based answer extraction model which effectively incorporates FrameNet style role semantic information and achieves promising results. The authors demonstrate performance gains over a shallow semantic parser trained on the FrameNet annotated corpus. Those and other research works showed that SRL gives promising result for different NLP applications.

Current approaches to semantic role labeling are based on supervised machine learning, often using the FrameNet and PropBank resources to specify what counts as a predicate, define the set of roles used in the task, and provide training and test sets.

1.2 Motivation

SRL has received considerable interest in the past few years [10, 11] because of its significant contribution for different natural language understanding applications such as information extraction, question answering, machine translation, summarization, co-reference resolution, etc. Currently such types of natural language understanding applications were developed for different languages including Amharic using different techniques. When SRL is used as an intermediate step for such application, it significantly improves their performances. However, still now there is no automatic Amharic SRL due to its difficulty to adopt SRL of other languages for Amharic language since Amharic language is morphologically complex and the general role of language specific characteristics in the extraction of semantic content is different. Thus, this situation has motivated us to develop automatic SRL for Amharic text.

1.3 Statement of the Problem

Natural language processing community has recently experienced a growth of interest in semantic roles. One of the foundational works on semantic role labeling was done by Gildea and Jurafsky [10] for English. Pradhan *et al.* [11] also propose a machine learning algorithm for shallow semantic parsing for English by extending the work of Gildea and Jurafsky [10]. Sun and Jurafsky [12] also present Shallow Semantic Parsing of Chinese.

To the best of our knowledge, there is no prior work on automatic SRL for Amharic. Although a number of algorithms have been proposed for automatically assigning semantic roles to other languages, these algorithms could not be directly applied to the Amharic language due to the difference in the general role of language specific idiosyncrasies in the extraction of semantic content and morphological complexity of Amharic language. Hence, this study will fulfill the gap and address the issue of automatic sematic role labeler for Amharic text using memory based learning.

1.4 Objectives

General Objective

This general objective of this thesis work is to design automatic semantic role labeler for Amharic text.

Specific Objectives

The study will have the following specific objectives:

- Review literature on the techniques of automatic semantic role labeling.
- Select appropriate model and algorithm for Amharic SRL.
- Adapt architecture for SRL.
- Prepare semantically annotated corpus to train and test the system.
- Develop prototype of the system.
- Test the performance of the system.

1.5 Methods

Literature Review

Detail review and assessment will be conducted through works which have been done on the area of semantic role labeling technique and related issues. In addition, areas including syntactic analysis and morphological analysis will be reviewed since they are building blocks of a natural language.

Data Collection

The data used to prepare the corpus will be collected from various sources of Amharic text.

Prototype Development

In order to evaluate the performance of the proposed solution a prototype system will be developed to automatically generate semantic role for Amharic sentence constituents.

Evaluation Metrics

The proposed system will be evaluated using relative metrics such as precision, recall, accuracy and F-score that help to properly measure the performance of the proposed system about how it solves the identified problem.

1.6 Scope and Limitations

The scope of this research work is limited to develop SRL for simple Amharic sentences.
We faced with two main constraints. These are the number of sample simple sentences were difficult to annotate easily for us without professional support. In addition, luck of adequate documents on the background development of automatic SRL for Amharic texts due to the scarcity of local reference materials near the study area.

1.7 Application of Results

The result of this research work can be one of the intermediate component of higher level NLP applications. A system has crucial roles in many areas of NLP for Amharic language. Therefore, researchers who are involved in increasing the capability of computers in processing Amharic language may benefit from the result of this study. Specially, researchers in the area of machine translation, information extraction, question answering and text summarization are among the main beneficiaries. One of the major performance bottlenecks in such type of applications are understands semantic content of given text. Therefore SRL significantly improve their performance by simplify this problem.

1.8 Organization of the Rest of the Thesis

The remaining parts of this thesis are organized as follows. Chapter two presents the review of literature in the area. Chapter three presents related works done on sematic role labeling. The fourth chapter is about the design of Amharic SRL. Chapter five presents the evaluation of Amharic SRL for simple Amharic sentences. Finally conclusions and future work are given in Chapter six.

CHAPTER TWO: LITERATURE REVIEW

2.1 Natural Language Processing

Natural language processing is a new interdisciplinary field variously called computer speech and language processing or human language technology or computational linguistics. It is the computerized approach to analyze text based on both set of theories and set of technologies. The goal of this field is to make computers perform useful tasks involving human language, such as enabling human-machine communication, improving human-human communication, or simply doing useful processing of text or speech [3]. Studying the complex language behavior needs good understanding of the language (level of linguistic analysis). For instance,

- **Phonetics and Phonology:** Knowledge about linguistic sounds.
- **Morphology:** Knowledge of the meaningful components of words.
- **Lexical:** Knowledge about lexical meaning of words and parts of speech analysis.
- **Syntax:** Knowledge of the structural relationships between words (grammar and structure of sentence).
- **Semantics:** Knowledge of meaning of a word, phrase and sentence.
- **Pragmatics:** Concerned with the purposeful use of language in a situation and utilizes context over and above the contents of the text for understanding.
- **Discourse:** Knowledge about linguistic units larger than a single utterance, i.e., deals with the properties of the text as a whole that convey meaning by making connection between component sentences.
- **Disambiguation:** Refers to the resolution of ambiguities that occur at different levels of language analysis.

A natural language processing task may involve all or some of these levels of analysis. Our study focuses on the area of semantic. In particular, we will deal with one type of semantic annotation that is called semantic role labeling. The need for meaning representations arises when neither the raw linguistic inputs nor any of the structures derived from them by any of the transducer facilitate the kind of semantic processing that is desired. More specifically, we need representation that bridges the gap from linguistic input to the non-linguistic knowledge of the

world needs to perform tasks involving the meaning of linguistic inputs [3]. Therefore, semantic role labeling is a good way toward this goal.

2.2 Semantic Role

Semantic role is the relationship between a syntactic constituent (verb's argument) and a predicate. It identifies the role of a verbal argument in the event expressed by the verb: an agent, a patient, an instrument, etc. Semantic roles are abstract models of the role an argument plays in the event described by the predicate [3]. This can be discussed in three different levels of generality:

- Verb-specific semantic roles

  E.g. runner, killer, speaker, broker, etc.
- Thematic relations: which are generalizations across the verb-specific roles.

  E.g. agent, instrument, experiencer, theme, patient
- Generalized Semantic Roles: are generalizations across thematic relations. Two arguments are used: actor and undergoer. Actor is a generalization across agent, experiencer, instrument and other roles. Undergoer is a generalization subsuming patient, theme, recipient and other roles.

Semantic role gives us a way to express some of the semantic of an argument in its relation to the predicate. Semantic roles correspond, grossly, to the well-known notion of lead: "WHO did WHAT to WHOM, HOW, WHEN and WHERE" [15].

Semantic role labeling is the task of automatically finding the semantic roles of each argument of each predicate in a sentence [3]. It takes the initial steps in extracting meaning from text by giving generic labels or roles to the words of the text. Typically, the role labeling task consists of identifying the constituents of each target predicate (argument identification) and labeling them with semantic roles (argument classification). Nevertheless, in order to identify and then classify these arguments, SRL systems first have to identify the target predicate (predicate identification) and then assign a certain sense number to it (predicate classification). The input information contains several levels of annotation apart from the role labeling information: POS tags, chunks, named entities, and parse trees [3].

2.2.1 Common List of Semantic Roles

Arguments are grouped into two major types according to their semantic roles: (a) necessary arguments, representing central participants in an event, which include agent, patient, instrument, etc.; (b) optional arguments (adjuncts), optional for an event but supplying more information about the event, which includes location, time, manner, etc. In this section, we will see a list of the major thematic relations usually considered based on [4].

Agent: Initiator of action, capable of volition, deliberately performs the action.

Example: [ተስፋ ድርጅት] *Agent* በጋምቤላ 480 ወላጅ አልባ ህጻናትን እረዳ

Patient: Affected by action, undergoes the action and has its state changed.

Example: ሰራተኞቹ [አጥሩን] *Patient* አፈረሱት

Experiencer: Entity moving or being "located" receives sensory or emotional input.

Example: የህዳሴ ግንብ ግንባታ [የኢትዮጵያን ህዝብ] *Experiencer* አስደሰተ

Theme: Entity moving, or being "located", undergoes the action but does not change its state. Sometimes used interchangeably with patient.

Example: ማእከሉ [67 ምርምሮችን] *Theme* አጠናቀቀ

Instrument: Used to carry out the action.

Example: ካሳ [በቁልፍ] *Instrument* በሩን ከፈተ

Location: Where the action occurs.

Example: ከ15 ሺ ኩንታል በላይ የቅመማ ቅመም ምርት [በጋሞጎፋ] *Location* ተገኘ

Direction or Goal: Where the action is directed towards.

Example: ድርጅቱ ምርቱን [ወደ መካከለኛው ምስራቅ ሀገራት]*Direction* ላከ

Source: Where the action originated.

Example: ገበሬው [ከማሳው] *Source* ወደ ከተማ መጣ

Time: The time at which the action occurs.

Example: የጎንደር ከተማ ዘላቂ የመጠጥ ውሃ ፕሮጀክት [ዛሬ] *Time* ተመረቀ

Beneficiary: The entity for whose benefit the action occurs.

Example: አዲስ አበባ ዩኒቨርሲቲ [ለዶክተር ብርሃነ] *Beneficiary* የፕሮፌሰርነት ማእረግ ሰጠ

Manner: The way in which an action is carried out.

Example: በትግራይ ደቡባዊ ዞን የቅድመ ጋብቻ ምርመራ የሚያደርጉ ተጋቢዎች ቁጥር [በ ፊጥነት] *Manner* ጨመረ.

Purpose: The reason for which an action is performed

Example: ጤና ጥበቃ [የወባ በሽታ ተጠቂዎችን ቁጥር ለመቀነስ] *Purpose* እንቅስቃሴ ጀመረ.

Cause: What caused the action to occur in the first place; not for what, rather because of what?

Example: [በደረቅ ቼክ ሳቢያ] *Cause* የ142 ተጠቃሚዎች ሂሳብ ተዘጋ.

2.2.2 Challenge in Semantic Role Labeling

Semantic role assignment is crucial to develop an efficient NLP system. But, despite this potential benefit, it has proved very difficult to come up with a standard set of roles, and equally difficult to produce a formal definition of roles like AGENT, THEME, or INSTRUMENT [16]. Consider the AGENT role; most cases of AGENTS are animate, volitional, sentient, causal, but an individual noun phrase might not exhibit all of these properties.

2.3 Component for Sematic Role Labeler

Automatic processing of semantic roles is a relatively new subject of study [16]. It relies strongly on the experience and results achieved in other domains of automatic text analysis such as morphological and syntactic, which provide linguistic knowledge about the text that is necessary for this task.

2.3.1 Morphological Analyzer

The term morphology comes from Greek word, morph- means ‘shape, form’, and morphology is the study of form or forms of something depends on which area it is used. When the word comes to linguistics, it means the study of the formation of words and their internal structure [17].

Morphological analysis is the process of finding the morphemes of the word and providing grammatical information for the word based on the identified morphemes. Morpheme is the smallest unit of a language that carries meanings. It could not be broken into smaller meaningful

unit. For example, the word "ሄደች/*hedäčə*" has two morphemes; "ሄደ/*hedä*" (masculine gender) and "-ች/*'čə*" (feminine gender) [18]. A program or system that performs morphological analysis is known as morphological analyzer.

Dealing with morphology of a language is analyzing different structure of the word for further analysis. It is crucial for many higher level natural language applications such as machine translation, speech recognition, spelling checker, semantic analysis, etc. Specifically toward the development of SRL for Amharic text it helps to correctly extract different features from constituents such as grammatical voice of the verb (active, passive).

2.3.2 Part of Speech Tagger

Part of speech tagging (POS tagging or POS), also called grammatical tagging or word category disambiguation, is a technique of assigning each word of a written text with an appropriate parts of speech tag. A part of speech is a category of words which have similar grammatical properties. Words that are assigned to the same word part of speech generally display similar behavior in terms of syntax. They play similar roles within the grammatical structure of sentences and sometimes in terms of morphology, in that they undergo inflection for similar properties [3].

POS tagging can be used as an intermediate step for higher level NLP tasks such as parsing, Text to Speech (TTS), Information Retrieval (IR), shallow parsing, Information Extraction (IE) semantic analysis, machine translation, etc. POS tagging, thus, is a necessary application for advanced NLP applications [3].

2.3.3 Parser

Parsing (syntax analysis or syntactic analysis**)** defined as the process of identifying the structure of specific sentence namely noun phrase (NP), verb phrases (VP), noun (N), verb (V) etc. according to a given grammar [19]. It also deals with a number of sub problems such as identifying constituents that can fit together, testing the compatibility of number and tense [19].

Parsing of sentences is believed to be an important task on the road to natural language understanding, and has immediate applications in tasks such as phrase recognition, conceptual parsing, machine translation, question answering, spell checker, text summarization, etc. [19]. It is also a crucial application for SRL in order to easily identify all constituents of sentences.

Existing parsers can be divided into two types according to their annotation schema phrase-structure and dependency structure.

Phrase structure focuses on identifying phrases and their recursive structure. X-Bar theory is used to flesh phrase structure rule and it is part of Chomskian linguistics. Figure 2.1 [20] illustrate X-bar theory. It is widely regarded as a substantive theory of phrase structure properties in natural languages.

X-bar Theory

- *Specifier Rule: XP→(YP) X '*
- *Modifiers Rule: X '→(ZP) X '*
- *Complement Rule: X '→ (WP) X*

Terminology

- *Specifier (YP): daughter of XP, sister to X '.*
- *Adjunct (ZP): daughter of X ', sister to X '.*
- *Complement (WP): daughter of X ', sister to X.*

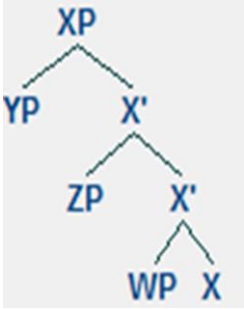

Figure 2.1: X-bar Theory

Dependency structure can generally be regarded as triples consisting of a head word, relation and a dependent [5]. E.g., in the sentence “ካሳ ምሳውን በላ”, “ካሳ” has a subject relation with the predicate “በላ” and “ምሳውን” has an object relation with the predicate “በላ”. This relation can be written as the triple (ካሳ, subject, በላ). Dependency trees can be represented with arrows pointing from the head to the dependents or from the dependents to the heads. Figure 2.2 shows dependence structure for the simple Amharic sentence “ካሳ ምሳውን በላ /Kassa məsawənə bäla/ Kassa ate his lunch”.

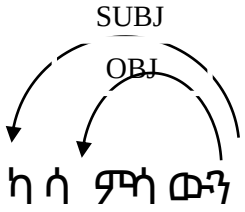

Figure 2.2: Dependency Tree for an Amharic Sentence

- **Heads**: It is the phrasal head of the encapsulating syntactic constituent. For example, the head of a noun phrase node is a noun. Every node has at most one head daughter.
- **Complements**: The way the thematic structure of the head must be interpreted is determined by its complements. Examples of complements are subject, direct object, indirect object, as well as arguments with a less obvious thematic role such as verbal complements. A node can only have one complement daughter of every category, e.g., one subject, one direct object, etc.
- **Modifiers**: Modifiers mark such notions as time, place and quantity. Modifiers can be omitted without affecting the thematic structure. A node can have several modifier daughters.

2.4 Lexical Resource

This section looks at the useful lexical resources used for semantic role labeling. These prominent lexical resources for semantic role labeling are FrameNet and PropBank.

2.4.1 FrameNet

FrameNet is an electronic resource and a framework for explicit description of the lexical semantics of words. It is intended to be used by lexicographers, but also by systems for natural languages processing [16]. The key concept in the FrameNet method of annotation is a semantic frame. A semantic frame can be described as a representation of an object, event or situation. Each frame has its own set of role. For example, the roles defined for the frame research are field, question, researcher and topic. Frames are evoked by verbs that are semantically related to the frame. For example, the frame research is evoked by investigation and research. Roles in FrameNet are called frame elements (FEs), the frame-evoking words are called lexical units (LUs) [5].

Each semantic frame in FrameNet is defined with respect to its frame elements, which are fine-grained semantic role labels. A frame is a structure used to define the semantic meaning of a word. It is a generalizable concept with recurring frame elements that are recognized intuitively. Frame elements are the elements which make up a frame. In FrameNet dataset, the sentences are

arranged in a hierarchical order with each frame referring to a concept. Frames at the higher level refer to a more generic concept while frames at the lower level refer to more specific concepts [2]. The frame elements for an individual frame are classified in terms of how central they are to a particular frame [21]. Three levels of semantic roles can be distinguished:

Core Elements: Conceptually necessary for the frame, roughly similar to syntactically obligatory, instantiate required roles, furthermore, core elements make the frame unique from other frames.

Peripheral Elements: Not central to the frame, but providing additional information about the event, such as time and place; roughly similar to modifiers.

Extra-thematic Elements: Not specific to the frame and not standard modifiers but describing the frame with respect to a broader context.

Every frame has invoking predicates attached to it. Figure 2.3 [10] shows structure of frames in the FrameNet lexicon. These are the verbs and some nouns that invoke the concept, referred by the frame they are attached this sentences that have these predicates would have constructs that play the role given by the frame elements of the invoked frame. For example, [$_{judge}$ she] blames [$_{evaluee}$ the government] [$_{reason}$ for failing to do enough to help]; in this example predicate blame invokes the judgment frame and other constructs in the sentence play the invoked semantic roles. (She) plays the role (Judge), (the Government) plays the role (Evaluee), (for failing to do enough to help) plays the role (Reason) [5].

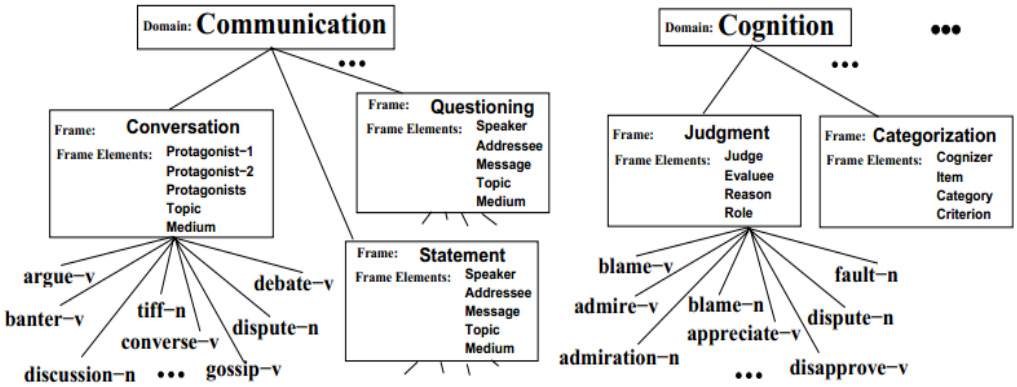

Figure 2.3: Sample Domain and Frames from the FrameNet Lexicon

In general, FrameNet is focused on semantic frames, which are defined as a schematic representation of situations involving various participants, props, and other conceptual roles [5]. The project methodology has proceeded on a frame-by-frame basis, that is by first choosing a semantic frame (e.g., Commerce), defining the frame and its participants or frame elements (Buyer, Goods, Seller, Money), listing the various lexical predicates which invoke the frame: buy, sell, etc., and then finding example sentences of each predicate in a corpus and annotating each frame element in each sentence. Parse trees are not used in FrameNet; annotators mark the beginning and end points of frame elements in the text, and add a grammatical function tag expressing the frame element's syntactic relation to the predicate.

2.4.2 PropBank

The Proposition Bank, generally referred to as PropBank, is a resource of sentences annotated with semantic roles [3]. PropBank is a corpus of naturally occurring sentences with manually annotated syntactic structure and semantic roles [22]. It is constructed by assigning semantic arguments to constituents of the hand-corrected TreeBank parses [5]. It is intended to be used for developing systems for natural language understanding that depends on semantic parsing, but also for quantitative analysis of syntactic alternations and transformations [16]. The labels for the semantic roles were attached to the corresponding nodes in the syntactic trees. The syntax tree representation along with the argument labels for the sentence “It operates stores mostly in Iowa and Nebraska” is shown in Figure 2.4 [22]. Proposition Bank project takes a practical approach to semantic representation, adding a layer of predicate-argument information, or semantic role labels, to the syntactic structures of the Penn Treebank [22].

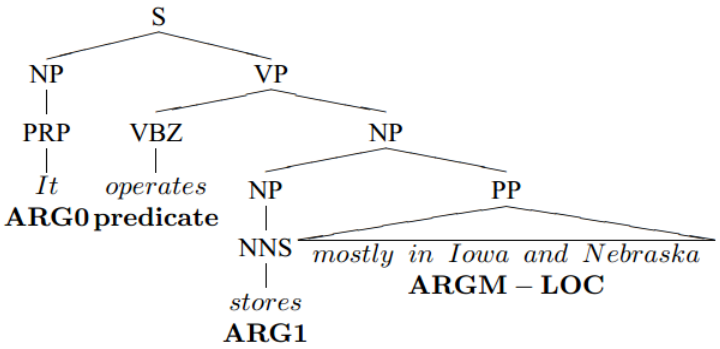

Figure 2.4: Syntax Tree for a Sentence Illustrating the Propbank Tags

Because of the difficulty of defining a universal set of semantic or thematic roles covering all types of predicates, PropBank defines semantic roles on a verb by verb basis [12]. Table 2.1 [22] illustrates argument labels associated with the predicate operate in the PropBank corpus. The semantic arguments of an individual verb are numbered from 0 to 4. For a particular verb, ARG0 is generally the argument exhibiting features of an Agent, while ARG1 is a Patient or Theme. Arguments that are labeled from ARG0 to ARG4 are called core arguments. Those numbered labels represent semantic roles of a very general kind.

Table 2.1: Argument Labels Associated with the Predicate Operate in the Propbank Corpus

No	**Tag**	**Description**
1	ARG0	Agent, operator (Agent, causer, experiencer)
2	ARG1	Patient, Theme
3	ARG2	Beneficiary/ Experiencer / Instrument / Attribute / End state
4	ARG3	Starting point / Beneficiary / Instrument / Attribute
5	ARG4	Ending point

In addition to verb-specific numbered roles, PropBank defines several more general roles that can apply to any verb. These adjuncts (circumstantial objects) are marked as ARG-Ms (modifiers). They can appear in any verb's frame, valence-independently. There are 12 secondary tags for ARGMs in the Proposition Bank: DIR, LOC, MNR, TMP, EXT, REC, PRD, PRP, DIS, ADV, MOD, NEG based on [24]. Table 2.2 [24] gives the list of PropBank annotated adjuncts.

Table 2.2: Propbank List of Annotated Adjuncts with their Explanations

ARGM type	Description
DIR	Direction
LOC	Location
MNR	Manner
TMP	Time
EXT	Extent
REC	Reciprocal
PRD	Secondary predication
PRP	Purpose
CAU	Cause
DIS	Discourse connectives
ADV	General-purpose adverbs
MOD	Modal verb
NEG	Negation marker

The PropBank numbered arguments are meant to be interpreted in a predicate specific manner whereas the ARGM'S have a global interpretation.

General procedure to design PropBank is based on creating frame sets for verbs and then using them as annotation guidelines for the annotation. As described in [22] the PropBank development process is divided into two parts: framing and annotation.

Framing

Framing is the process of creating the frames files, that is, the collection of frame set entries for a verb. The frame sets identify the predicate and its possible arguments. In PropBank, frame files provide a verb specific description of all possible semantic roles and we illustrate these roles by examples. For example, frame set for the verb ኣጸደቐ/ʾäs'ädäqä (approved):

Roles:

- ARG0: approver
- ARG1: thing approved

Example: Transitive, active:

ምክር ቤቱ ሁለት ረቂቅ አዋጆችን አጸደቀ /məkərə betu hulätə räqiqə ʾäwağočənəmə ʾäs'ädäqä (The Council approved two draft edicts)

- ARG0: *ምክር ቤቱ*
- REL: *አጸደቀ*
- ARG1: *ሁለት ረቂቅ አዋጆችን*

Some verbs may have different sense at this time. It is impossible to provide one set of semantic roles for all verb sense. For example, in the following two sentences, the verb "ሳለ" takes different argument. In such cases, frame files distinguish two or more verb senses, which are called Frame sets, and define argument labels specific to each frame set:

– ካሳ (ARG0) ጉንፋን ስለያዘው (ARGM-PUR) ሳለ
– ካሳ (ARG0) ስዕል (ARG1) ሳለ

Frame files are essential for consistent annotation, especially when a corpus is annotated by many annotators. Moreover, the lack of example sentences makes consistent annotation even more difficult.

Annotation

Choosing ARG0 versus ARG1

The ARG0 label is assigned to arguments which are understood as agents, causers, or experiencers. The ARG1 label is usually assigned to the patient argument, i.e., the argument which undergoes the change of state or is being affected by the action.

ARG0 arguments are the subjects of transitive verbs and a class of intransitive verbs called unergatives.

– ካሳ (ARG0) ምሳጩን በላ
– ካሳ (ARG0) ተኛ

Semantically external arguments have what are called Proto-Agent properties such as [23]:

- Volitional involvement in the event or state
- Causing an event or change of state in another participant
- Movement relative to the position of another participant

Internal arguments (labeled as ARG1) are the objects of transitive verbs and the subjects of intransitive verbs called unaccusatives:

  - ካሳ ሞስኮቱን (ARG1) ሰበረው
  - ሞስኮቱ (ARG1) ተሰበረ

These arguments have Proto-Patient properties, which means that these arguments

- Undergo change of state
- Are causally affected by another participant
- Are stationary relative to movement of another participant

Whereas for many verbs, the choice between ARG0 and ARG1 does not present any difficulties, there is a class of intransitive verbs (known as verbs of variable behavior), where the argument can be tagged as either ARG0 or ARG1 [25, 26].

Arguments which are interpreted as agents should always be marked as ARG0, independent of whether they are also the ones which undergo the action. In general, if an argument satisfies two roles, the highest ranked argument label should be selected (ARG0 >> ARG1 >> ARG2>>, etc.) [25, 26]. Given this rule, agents are ranked higher than patients. If an argument is both an agent and a patient, then ARG0 label should be selected. Not all ARG0s are agentive. However, there are many inanimate as well as clausal arguments which are being marked as ARG0s. These arguments are usually the ones which cause an action or a change of state [25, 26].

Annotation of null elements

Passive sentences

Sentences can be either active or passive. In active sentences, the subject is the agent or a doer of the action, marked as ARG0 in Propbank. In passive sentences, the subject of the sentence is acted upon by some other agent or by something unnamed, and is being marked as ARG1 in Propbank [25, 26].

Passive sentences are derived from the corresponding active sentences by 'movement' of the object to the subject position [25, 26]. This movement leaves a trace, represented as [*T*] in Treebank. For example:

- Active: Mary hit John
- Passive: John was hit [*T*] by Mary.

Propbank annotation:

- REL: hit
- ARG1: [*T*] -> John
- ARG0: by Mary

Special cases

Verbs of Saying

A verb of saying is any verb which has a speaker argument (ARG0) and an utterance (ARG1). If the utterance argument can be selected as one constituent, then ARG1 is a single constituent or a chain [25, 26]. For example: Mary said we will win.

Propbank Annotation

- REL: said
- ARG1: [*T*-1] -> we will win
- ARG0: Mary

Annotation Modifier

Comitatives (COM)

Comitative modifiers indicate who an action was done with. This can include people or organizations (Entities that have characteristics of prototypical agents: Animacy, volition) but excludes objects, which would be considered instrumental modifiers [25, 26].

E.g. ካሳ ከ እህቱ ጋር ሙዚቃ አቀረበ / Kassa kä ʾəhətu garə muziqa ʾäqäräbä

- ARG0: ካሳ
- REL: አቀረበ
- ARG1: ሙዚቃ
- ARGM-COM: ከ እህቱ ጋር

Locatives (LOC)

Locative modifiers indicate where some action takes place [25, 26].

E.g. ካሳ ከ እህቱ ጋር በ ካፒታል ሆቴል ሙዚቃ አቀረበ / Kassa kä ʾəhətu garə bä kapitalə hotelə muziqa ʾäqäräbä

- ARG0: ካሳ
- REL: አቀረበ
- ARG1: ሙዚቃ
- ARGM- LOC: በ ካፒታል ሆቴል
- ARGM-COM: ከ እህቱ ጋር

Destination (DES)

Destination modifier indicates the final resting place or destination of motion. However, if there is no clear path being followed, a ‘location’ marker should be used instead [25, 26].

E.g. ስደተኞቹ ወደ ሀገራቸው ተመለሱ/ sədätäñoču wädä hägäračäwə tämäläsu

- ARG1: ስደተኞቹ

- REL: ተመለሱ
- ARGM-DES: ወደ ሀገራቸው

Manner (MNR)

Manner adverbs specify how an action is performed. For example, ‘works well’ is a manner. Manner tags should be used when an adverb could be an answer to a question starting with ‘how?’ [25, 26].

E.g. ወፎቹ ስንዴውን በችኮላ ለቀሙ

- ARG0: ወፎቹ
- REL: ለቀሙ
- ARG1: ስንዴውን
- ARGM-MNR: በችኮላ

Temporal (TMP)

Temporal ARGMs show when an action took place, such as ‘in 1987’, ‘last Wednesday’, ‘soon’ or ‘immediately’. Also included in this category are adverbs of frequency e.g., often, always, sometimes (with the exception of ‘never’, see NEG below), adverbs of duration (for a year/in an year), order (e.g., first), and repetition (e.g., again)[25, 26].

E.g. ካሳ ትናንትና ከእህቱ ጋር በ ካፒታል ሆቴል ሙዚቃ አቀረበ

- ARG0: ካሳ
- ARGM- TMP: ትናንትና
- REL: አቀረበ
- ARG1: ሙዚቃ
- ARGM- LOC: በ ካፒታል ሆቴል
- ARGM-COM: ከ እህቱ ጋር

Extent (EXT)

ARGM-EXT indicate the amount of change occurring from an action, and are used mostly for the following[25, 26]:

- Numerical adjuncts like‘ (raised prices) by 15%’,
- Quantifiers such as ብዙ፣ትንሽ
- And comparatives such as ‘(he raised prices) more than she did’

E.g. የጤፍ ዋጋ ከአምና በ 10% ጨምሯል

- ARG0: የጤፍ ዋጋ
- REL: ጨምሯል
- ARGM-EXT: በ 10%
- ARGM- TMP: ከአምና

Secondary Predication (PRD)

PRD modifies another argument of the verb (describing its state during or after the event) more than it modifies the verb or event itself.

E.g. ሮናልዶ የእግር ካስ ቡድኑ አንበል ሆነ

- ARG0: ሮናልዶ
- REL: ሆነ
- ARGM-EXT: የእግር ካስ ቡድኑ አንበል

Purpose Clauses (PUP)

Purpose clauses are used to show the motivation for some action. Clauses beginning with ‘in order to’ and ‘so that’ are canonical purpose clauses.

E.g. ለባለሙያዎች ስልጠና 43 ሚሊየን ብር ተመደበ

- ARGM-PUR: ለባለሙያዎች <NP> ስልጠና <N>
- ARG1-TEM: 43 ሚሊየን <NUM> ብር <N>
- REL:ተመደበ <V>

Cause Clauses (CAU)

Similar to ‘Purpose clauses’, these indicate the reason for an action. Clauses beginning with ‘because’ or ‘due to’ are canonical cause clauses. Questions starting with ‘why’ and ‘which’ are always treated as a cause. However, in these question phrases it can often be difficult or impossible to determine if the ‘why’ truly represents purpose or cause. Thus, as a general rule, if the annotator cannot determine whether an argument is more appropriately purpose or cause, cause is the default choice.

Negation (NEG)

Negation is an important notation for PropBank annotation. Therefore, all markers which indicate negation should be marked as NEG. Most of the time in Amharic sentences negation marker are indicated using prefix “አይ”፣“አት” such as አይመጣም ፣ አይበላም ፣ አትመጣም ፣ አትበላም. Those are identified using morphological analyzer.

Both FrameNet and PropBank resources used to specify what counts as a predicate, define the set of roles used in the task, and provide training and test sets. Recall that the difference between these two models of semantic roles is that FrameNet employs many frame-specific frame elements as roles, while PropBank uses a smaller number of numbered argument labels that can be interpreted as verb-specific labels, along with the more general ARGM labels.

2.5 Approaches to Semantic Role Labeling

In the last few years, there have been a lot of interest and activity in developing new approaches to learning for natural language processing [27]. Various learning methods have been used including Rule-based, Statistical or Connectionist and various hybrid approaches. Rule-based approaches such as Head-Driven Phrase Structure Grammar (HDPSG) have their shortcomings: they are time-consuming and have limited coverage. Most of the SRL approaches are statistical, making use of a variety of models [2].

Rule-based Approach: This approach has successfully been used in developing many natural language processing systems [28]. It is based on explicit representation of facts about language

through well-understood knowledge representation schemes and associated algorithms. It usually consists of a set of rules, an inference engine, and a workspace or working memory. Knowledge is represented as facts or rules in the rule-based approach. The inference engine repeatedly selects a rule whose condition is satisfied and executes the rule. The primary source of evidence in rule-based systems comes from human-developed rules (e.g. grammatical rules) and lexicons.

The advantage of the rule-based approach over the corpus-based approach is for: 1) Less-resourced languages, for which large corpora, possibly parallel or bilingual, with representative structures and entities are neither available nor easily affordable, and 2) For morphologically rich languages, which even with the availability of corpora suffer from data sparseness [28].

Statistical Approaches: Statistical NLP comprises all quantitative approaches to automated language processing, including probabilistic modeling, information theory, and linear algebra [29]. It often uses large text corpora to develop approximate generalized models of linguistic phenomena based on actual examples of these phenomena provided by the text corpora without adding significant linguistic or world knowledge.

In statistical NLP, people cannot actually work from observing a large amount of language use situated within its context in the world. So, instead, people simply use texts, and regard the textual context as a surrogate for situating language in a real world context. A body of texts is called a corpus. Corpus is simply Latin for 'body,' several collections of texts formulate a corpora. They are usually characterized by such large text corpora and performing some analysis which uses primarily the text characteristics without adding significant linguistic or world knowledge [27].

A connectionist Approach: is a network of interconnected simple processing units with knowledge stored in the weights of the connections between units. Similar to the statistical approaches, connectionist approaches also develop generalized models from examples of linguistic phenomena. What separates connectionism from other statistical methods is that connectionist models combine statistical learning with various theories of representation. In addition, in connectionist systems, linguistic models are harder to observe due to the fact that connectionist architectures are less constrained than statistical ones [27].

Learning methods are designed to support automated knowledge acquisition, fault tolerance, and plausible induction. Using learning methods for natural language processing is especially important because learning is an enabling technology for many language-related tasks, such as speech recognition, spoken language understanding, machine translation, and information retrieval. Furthermore, learning is important for building more flexible, scalable, adaptable, and portable natural language systems [27].

The first two approaches rule-based and connectionist are most commonly used by different researchers to develop SRL success has also been achieved by statistical techniques [2]. Most of the current statistical approaches to SRL are supervised machine learning technique, requiring large quantities of human annotated data to estimate model parameters [6]. SRL is commonly developed using a supervised learning paradigm. In the next section we will see in detail about supervised machine learning technique, unsupervised approach and semi-supervised approach for SRL.

2.5.1 Supervised Approach to Semantic Role Labeling

The most widely used procedure for automatic labeling of semantic roles is the supervised machine learning technique defined in [3].Current approach to semantic role labeling rely on role-annotated data such as FrameNet and PropBank resources to specify what counts as a predicate, define the set of roles used in the task, and provide training and test sets [3]. Supervised systems offer a more flexible approach to role labeling. By training them on different sets of data they can be easily adapted to different text domains and even to different languages [5].

As described in [10] approaches in supervised machine learning consist of three major steps. In the first step, the system is trained on the text where segments of texts are already correctly labeled (with semantic roles in this task). It reads the text (training input) and collects the knowledge about the occurrences of labels. In the second step, the system reads a new text for which the labels are to be automatically assigned (test input) and attempts to predict the correct label for every given segment of the text using the information available in the text and the knowledge acquired in training. In the third step, the performance of the system is evaluated.

2.5.2 Unsupervised Approach to Semantic Role Labeling

Unsupervised learning is also referred to as rule-based method. It tries to utilize regularities in unlabeled data and thus make predictions without any manually annotated training data, i.e., it exploits unlabeled data for semantic role labeling. It groups instances together with some accuracy, but not to associate them with class labels as defined by the task, e.g., semantic role labels. This connection has still to be made either with the help of information from a manually created lexicon or by manually labeling a small set of instances.

Unsupervised semantic role labeling avoids the need for expensive manual labeling of text (does not make use of role-annotated data), and enables the use of a large, representative corpus [30]. Argument classification can be naturally formalized as a clustering problem where argument instances are assigned to clusters. Ideally, each cluster should contain arguments corresponding to a specific semantic role and each role should correspond to exactly one cluster. Unsupervised systems are especially useful when no training data is available.

2.5.3 Semi-Supervised Approach to Semantic Role Labeling

A major limiting factor for supervised approach is the need for large manually labeled semantic resources to train semantic role labeling systems. The existing semi-supervised approaches to SRL can largely be regarded as extensions to supervised techniques, as they use supervised learning as sub-routines in the estimation process. A semi-supervised approach to semantic role labeling requires only a small manually labeled corpus of role-annotated sentences, in addition to manually annotated corpora it also utilizes information gained from unlabeled corpora [13]. The semi-supervised learning paradigm assumes that manual creation of a small labeled resource of seed annotations is feasible, and that in addition to these labeled instances, there is a large amount of unlabeled instances.

The idea in semi-supervised learning is to alleviate the data requirements for semantic role labeling by extending existing resources through the use of unlabeled data [13].

2.6 Classification Methods for SRL

The most common type of machine learning algorithms applied to the SRL problem is classification algorithm. Classification can be defined as creating a mapping between a set of features and a set of predefined classes and is a popular technique in data mining [5]. Features are properties of the items to be classified, in the case of shallow semantic parsing information from a syntactic parse is used. Some commonly used features in SRL are listed and described in the next section. The goal of classification is to assign class labels to a set of instances automatically. Instances represent the items to be classified and their classes.

Classification algorithms are amongst the most popular methods of automatic role assignment (sometimes called shallow semantic parsing). In classification systems, text chunks are classified as a semantic argument or as a none-semantic argument (null argument). Semantic arguments can then be further classified into a set of argument labels. Often, classification is implemented as a supervised machine learning algorithm [5].

A wide range of classification algorithms have been developed and many of these have been applied to the SRL task in previous research. In this section, we will discuss Probability Estimation, Support Vector Machines (SVMs) and Memory based learning (MBL) classifiers which play a prominent role in SRL and analyze their strengths and weaknesses.

2.6.1 Probability Estimation

Probability Estimation is the first method applied to the SRL task. It is a classic statistical approach, first applied to semantic role classification by Gildea and Jurafsky [10]. The probability model considers the question of finding the boundaries of frame elements separately from the question of finding the correct label for a frame element, although similar features are used for both tasks. Given a constituent with feature vector $\vec{v}$, $P(r|\vec{v})$ is the probability that this constituent fills semantic role ***r***. For example, suppose we use the features ***gov*** (governing), ***voice*** and ***Pt*** (phrase type). Then, it would be possible to calculate the probabilities of the possible semantic roles by counting the number of times each role appears in combination with these features, and dividing that by the total number of times the combination of features appears:

$$p(r|gov, voice, pt) = \#(r, gov, voice, pt)/\#(gov, voice, pt) \quad (1)$$

A problem with this method is that if certain feature combinations are observed in the training data a small number of times, this method results in a poor probability estimate.

2.6.2 Support Vector Machines

Support Vector Machines (SVMs) are relatively new classification method performed well on text classification tasks [11], which is probably why many researchers have used them for semantic role classification. Proposed in 1992 by Boser et al. [11], SVMs are binary classifiers used for separating data into one of two categories. An N-dimensional hyper plane model is built from a training set. The training set consists of examples, each one labeled as belonging to one of two categories. The examples are mapped as points in a hyper plane, such that examples in separate categories are separated by the widest possible margin in the space. Once trained a new example will be mapped to the space and estimated to belong to the category on whichever side of the margin they have been mapped to. The ability of an SVM to learn is independent of the dimensionality of the hyper plane.

SVMs can handle an extremely large number of interacting or overlapping features with strong generalization properties and it performs well on the SRL task, but the training time can be considerable especially on large data sets. Furthermore, the theory behind SVMs is relatively complex, which makes them difficult to use [11].

2.6.3 Memory Based Learning

Memory Based Learning (MBL) is a supervised machine learning technique, based on the k-NN algorithm [5]. It is a direct descendant of the classical k-Nearest Neighbor (k-NN) approach. However, k-NN is used for classification of numeric data. Memory based learning can be described as reasoning on the basis of similarity of new situations to earlier encountered situations. The basic idea behind memory-based learning is that concepts can be classified by their similarity with previously seen concepts [31]. It has learning and performance components. Learning component of a MBL system is memory based: all training examples are stored in memory. The performance component is similarity-based and performs the actual

classification, i.e., Memory-based language processing is based on the idea that NLP problems can be solved by storing solved examples of the problem in their literal form in memory, and applying similarity-based reasoning on these examples in order to solve new ones [5]. Keeping literal forms in memory has been argued to provide a key advantage over abstracting methods in NLP that ignore exceptions and sub regularities.

During training, training instances are loaded into memory. An instance consists of a vector containing feature-value pairs and a class assignment. During classification, unseen examples are compared to instances in the training data. This comparison is done using a *distance metric.* The class assignment is based on the k-nearest neighbor algorithm: the most common class amongst the k most similar training instances is chosen. In case of a tie among categories, a tie breaking resolution method is used. Different distance metrics can be used in MBL. The most common of which is the overlap metric.

A k-NN algorithm with this metric is called IB1 [31]. In the overlap metric, all features have the same weight. To improve performance, domain knowledge can be used to assign different weights to different features. Weights can also be determined by calculating statistics (e.g., frequencies) of features in the training data. These statistics can be used to determine which features are good predictors of the class labels and which features are less relevant. A useful tool for measuring feature relevance that is especially beneficial for NLP tasks is information gain (IG). Information gain looks at each feature and measures how much information it contributes to the knowledge needed to predict the correct class label. The most common way of measuring the IG of a feature ***i*** is to compute the difference in uncertainty (i.e., entropy) between situations without and with knowledge of the value of that feature.

$$w_i = H(C) - \sum_{v \in V} p(v) * H(C|v) \quad \textbf{(2)}$$

where **C** is the set of class labels, $\mathbf{V_i}$ is the set of values for feature ***i,*** and **H(C)** is the entropy of the class labels. Entropy measures the amount of uncertainty of a variable, and is defined as:

$$H(X) = -\sum_{x \epsilon X} p(x) \log_2 p(x) \quad (3)$$

The probabilities are estimated from relative frequencies in the training set.

By using different data-structures and applying variety of speed optimizations, the Induction of Linguistic Knowledge (ILK) research group at Tilburg University adapted existing k-NN algorithms so that they can be used for natural language processing (NLP) applications. The result is TiMBL [31].

TiMBL is an open source software package implementing several memory-based learning algorithms such as IB1, IB2, IGTREE, TRIBL and TRIBL2 all implemented algorithms have in common that they store some representation of the training set explicitly in memory. During test, new cases are classified by extrapolation from the most similar stored cases. The main differences among the algorithms incorporated in TiMBL lie in the definition of similarity, the way the instances are stored in memory, and the way the search through memory is conducted [34,42]. Among which **IB1** is used in this study the generalized accuracy of **IB1** is relatively good that is why we are chosen **IB1. IB1** relies on the k -nearest-distances rule which states that all memory items which are equally near at the nearest distances surrounding the test item are taken into account in classification. Here the classification assigned to the test item is simply the majority class among the memory items at the nearest distances [31].

TiMBL input consists of text files with training and test instances. An instance is a tuple containing feature values and a target class. TiMBL supports several input formats, but they all have in common that every line in the input data file contains one instance and the last feature is considered the target class. For our work we are used the C4.5 (comma separated) format, which is the default format in TiMBL.

There are different optimized parameters to be tuned in TiMBL. These are the MVDM (modified value difference metric) from distance metrics, IG (information gain) from weighting metrics, ID (inverse distance) from class voting weights, and **k** value from the nearest neighbor. They are used together with the different classifiers (algorithms).

IBl

The classification function of IB1 computes the similarity between a new instance and all stored instances, and returns the class label of the most similar instance. It uses information gain weights in the overlap function to define similarity. In IBl the instance base is reorganized (by compression rather than by pruning) in such a way that access to relevant instances is faster, and no generalization performance is lost [32].

In IBl, Instance-based learning algorithm, similarity of a new case to stored cases is used to find the nearest neighbors of the new case. The class of the new case is predicted on the basis of the classes associated with the nearest-neighbor cases and the frequency of their occurrences. All features are assigned the same relevance. The similarity function in lazy learning consisted simply of multiplying, when comparing two feature vectors, the similarity between the values for each feature with the corresponding information gain, or in case of features with different numbers of values the gain ratio, for that feature [33].

2.7 Features Used in Assigning of Semantic Roles

Supervised learning algorithms need to be trained on a set of training data before they can be applied to unseen data, as opposed to the unsupervised methods. The most common type of machine learning algorithms applied to the SRL problem is classification algorithms. The goal of classification is to assign class labels to a set of instances automatically. Instances represent the items to be classified and their classes. Instances are usually encoded as a set of features. Each feature describes a different aspect of the item to be classified. Finding a feature set that yields an accurate classification is an important and certainly not trivial task [10].

The set of features used in SRL systems has grown over time as researchers have explored new ways of leveraging the syntactic analysis of the entire sentence to better analyze specific semantic roles. Thus, while early systems used only a handful of features, current state of the art systems use dozens. Nonetheless, the features used in the earliest systems continue to form the core of current SRL. In this section we will see features which have been used in predicting roles of individual constituents applied to FrameNet data by Gildea and Jurafsky [10] who are the developer of the first automatic semantic role labeler and features introduced in later systems also.

The following are core features that were applied for first time to semantic role assignment by Gildea and Jurafsk [10].

- **Phrase Type**: Reflects the fact that some semantic roles tend to be realized by one and others by another type of phrase. Different roles tend to be realized by different syntactic categories. For example, in FrameNet communication frames, the Speaker is likely to appear as a noun phrase, Topic as a prepositional phrase or noun phrase, and Medium as a prepositional phrase, as in: "[$_{\text{Speaker}}$ We] talked [$_{\text{Topic}}$ about the proposal] [$_{\text{Medium}}$ over the phone] ." Phrase types were derived automatically from parse trees generated by the parser.

- **Governing Category:** Defines the grammatical function of the constituent that realizes a particular semantic role. It is based on the fact that some semantic roles are realized as the subject in a sentence, and others are realized as the direct object. The feature called "governing category", or **gov**, has only two values, **S** and **VP**, corresponding to subjects and objects of verbs, respectively. This feature is restricted to apply only to **NPs**, as it was found to have little effect on other phrase types. If a constituent bearing a semantic role is governed by the node **S** in a syntactic tree, it is the subject of the sentence, if it is governed by the node **VP**; it means that it belongs to the verb phrase, which is the position of the object. The difference between the direct and the indirect object is not made. This feature restrictedly applied to **NPs** so if **NP** nodes found under **S** nodes are generally grammatical subjects, and if **NP** nodes under **VP** nodes are generally objects.

- **Parse Tree Path:** Like the governing category feature described above, this feature is designed to capture the syntactic relation of a constituent to the rest of the sentence. However, the path feature describes the syntactic relation between the target word (that is, the predicate invoking the semantic frame) and the constituent in question, whereas the previous feature is independent of where the target word appears in the sentence; that is, it identifies all subjects whether they are the subject of the target word or not. A parse tree path of a constituent of a sentence is the path of the constituent from the target word in the constituent parse tree which includes the intermediate nodes and the arrow directions. The path includes, as the first element of the string, the part of speech of the target word, and, as the last element, the phrase type or syntactic category of the sentence constituent marked as a frame element. After some experimentation, Gildea and Jurafsky [10] used a version of the path

feature that collapses the various part-of-speech tags for verbs, including past tense verb (VBD), third person singular present verb (VBZ), other present tense verb (VBP), and past participle (VBN), into a single verb tag denoted "VB." Figure 2.5 shows an example path between verb eat and noun phrase He as: VB ↑VP ↑ S ↓ NP

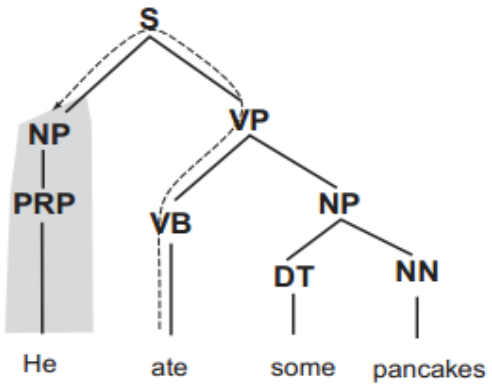

Figure 2.5: Parse Tree Path Example

For the purposes of choosing a frame element label for a constituent, the path feature is similar to the governing category feature defined above. Because the path captures more information, it may be more susceptible to parser errors and data sparseness.

- **Position**: In order to overcome errors due to incorrect parses, as well as to see how much can be done without parse trees, Gildea and Jurafsky [10] use position as a feature. This feature simply indicates whether the constituent to be labeled occurs before or after the predicate defining the semantic frame. This feature is highly correlated with grammatical function, since subjects will generally appear before a verb, and objects after.

- **Voice:** The distinction between active and passive verbs plays an important role in the connection between semantic role and grammatical function, since direct objects of active verbs often correspond in semantic role to subjects of passive verbs.

- **Head Word:** Lexical dependencies are extremely important in labeling semantic roles as indicated by their importance in related tasks such as parsing. Head words of noun phrases can be used to express selection restrictions on the semantic types of role fillers.

- **Sub-Categorization:** This feature refers to the set of a verb's syntactic arguments in the sentence. For example, we expect an intransitive use of close such as the door closed to have a different mapping from syntactic to semantic roles when compared to the transitive use in "He closed the door". The sub categorization feature of the first verb usage is {subject}, while the sub categorization for the second is {subject, object}.

- **Argument Set:** This feature, called Frame Element Group in the FrameNet-based system of Gildea and Jurafsky [10] is the set of all roles appearing for a verb in a given sentence. Since this feature depends on the roles assigned to all constituents in a sentence, it was employed in a post-processing ranking of role assignments for an entire sentence.

Features introduced by Pradhan *et al* [11]:

- **Argument Order**: This feature is an integer indicating the position of a constituent in the sequence of arguments for the given verb. It is computed after an initial phase classifies constituents as arguments or non-arguments. Because the feature does not use the syntactic parse tree, it can help make a semantic role labeling system robust to parser error.
- **Previous Role:** This feature is simply the label assigned by the system to the previous argument (skipping non-argument constituents), because this feature introduces a dependency among labels assigned to different constituents.

- **Head Word Part of Speech**: Because Penn Treebank part of speech categories distinguish singular from plural nouns, and proper and common nouns, the part of speech tags of the head of an NP can refine the type of noun phrase involved.

- **Named Entities in Constituents**: This feature uses the named entity recognizer to identify words as instances of the classes Person, Organization, Location, Percent, Money, Time, and Date. This feature helps handle the data sparsity caused by the unlimited sets of proper names for people, organizations, and locations in particular.
- **Verb Clustering:** Because semantically similar verbs such as eat and devour will occur with the same objects, they will be assigned to the same clusters. The use of this cluster

for predicting argument structure is motivated by the observation that semantically similar verbs undergo the same pattern of argument alternation.

- **Head Word of Objects of PPS:** When a verb's argument is a prepositional phrase (PP), the PP's head word is the preposition. While this can often be a reliable indicator of semantic role (for example in, across, and toward usually indicate location), most prepositions can be used in many different ways, and the sense can be determined by the preposition's object. For example, in February indicates time, while in New York indicates location.
- **First/Last Word/POS in Constituent**: As with the previous feature, this feature provides more specific information about the argument filler than the headword alone, but does so in a more general way that is robust to parser error and applies to any type of constituent.
- **Constituent Order**: This feature is related to the argument order feature above but is designed to be used in discriminating arguments from non-arguments. The position of each constituent is calculated relative to the predicate, which helps favor constituents close to the predicate.
- **Constituent Tree Distance:** This feature has the same motivation as the previous feature but takes syntactic structure into account.
- **Constituent Context Features**: These features provide information about the parent and left and right siblings of a constituent. For each of these three constituents, the phrase type, head word, and the head word's part of speech are given as features, for a total of nine features.
- **Temporal Cue Words:** A number of words which indicate time, but which are not considered named entities by the named entity tagger, are included as binary features.

2.8 Amharic Sentences

A sentence is a group of words or phrases. Phrase is a basic building block of a sentence. Phrase is a small group of words standing together. It is a syntactic structure that is wider than a word and smaller than a sentence. Phrase can be constructed only from a head word or a head word

combined with other words or phrases. When a phrase is constructed from a head word and other words or phrases, the other words or phrases can be specifiers, modifiers or complements.

Specifiers are words used to specify the identity, location, number, possession, etc. of the head word. Specifiers can be either primitive or derived. For example, in the phrase “የካሳ መጽሀፍ” (Kassa’s book) the specifier is “የካሳ” (Kassa’s) that shows the owner of the book.

Modifiers on the other hand are used to indicate the amount, time, place, type, etc. of the head word or phrase. Adjectival phrase, noun phrase, prepositional phrase or sentences can be considered as modifiers. For example, in the phrase “ነጭ መኪና” (white car), the word “ነጭ” (white) indicates what type of color the car has. Unlike specifiers and modifiers complements are words or phrases that are used to make the ideas complete. For example in the sentences “ዳቦ በላሁ” (I ate bread) and “የስንዴ ዳቦ በላሁ” (I ate wheat bread), the first sentence does not give complete information about the bread but in the second sentence “የስንዴ” (wheat) is a complement that indicates from what the bread is made.

The types of the phrases for a language are determined by the categories of words in a language. Therefore, since there are five word categories in Amharic language, there are five phrase classes which are noun phrase (NP), verb phrase (VP), adjectival phrase (AdjP), adverbial phrase (AdvP), and prepositional phrase (PP) [37].

Sentence is complete in itself, conveying a statement, question, exclamation, or command and typically containing a subject and predicate. For example, when we see the following phrases: “ አንድ ትልቅ ልጅ ʾänədə tələqə ləğə ” (One big child) - Noun phrase, “ በሁለት ብር bähulätə bərə ” (by two birr) - Prepositional phrase and “ አራት እንቁላል ገዛ ʾäratə ʾənəqulalə gäza” (he bought four eggs) - Verb phrase, they do not convey full meaning when they are spoken in separate, instead they raise questions like what did?, who did?, and what has done?. However, when these phrases are combined together they form a sentence “ አንድ ትልቅ ልጅ በሁለት ብር አራት እንቁላል ገዛ ʾänədə tələqə ləğə bähulätə bərə ʾäratə ʾənəqulalə gäza” (One big child bought four eggs by two birr) and answers all the above questions. From the structural point of view, sentences are basically constructed from noun phrase and verb phrase. Regarding the

remaining phrases, they may be included within one of the above main components of a sentence. Therefore, in the above sentence, we have the noun phrase “አንድ ትልቅ ልጅ” (one big child) and the verb phrase “በሁለት ብር አራት እንቁላል ገዛ” (bought four eggs by two birr) which in turn contains the prepositional phrase “በሁለት ብር” (by two birr) [18]. Figure 2.6 [18] shows a parse tree for the above example.

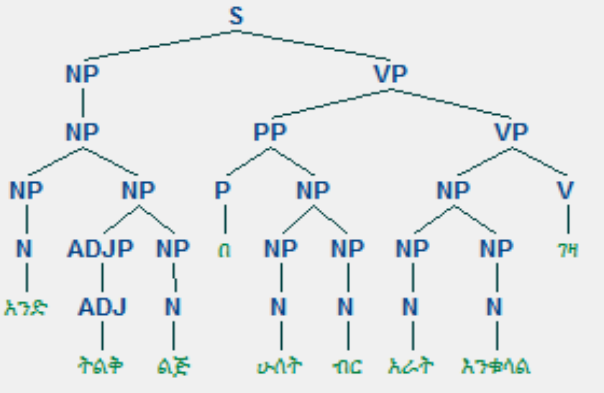

Figure 2.6: Parse Tree Structure for an Amharic Sentence

We call the sentence shown in Figure 2.6 simple because it only contains one verb. The noun phrase that forms sub verb phrase by combining with verb is called object without this the verb does not form phrase. The noun phrases that form the sentence by combining with verb phrase is called subject. When we see the parse tree, it is taken from the sentence node. So, it is different from other phrases which are taken from the verb phrases node. Generally, modifiers are taken from the verb phrase node and complements are taken from the sub-verb phrase. The subject and the object are mandatory. In parse tree structure based on nodes from which phrases are taken we can divide words by saying subject, modifier and complement [18].

The structure of a sentence can be either simple or complex based on the number of verbs it contains.

Simple sentence

A simple sentence consists only one main verb. For instance:

- አስቴር ብርጭቆውን ሰበረችው /äsəterə bərəč'əqowənə säbäräčəwə (Aster broke the glass)
- ካሳ ቢላዋውን በግ አረደበት /kasa bilawawənə bägə ʾärädäbätə (kasa slaughters the sheep with the knife)
- ካሳ ለወንድሙ ገንዘብ ላከለት/kasa läwänədəmu gänəzäbə lakälätə (kasa sends money to his brother)

Because each of the above sentences is constructed by using a single verb "ሰበረችው /ሰበር-ኧች-ው/ (breaks)፣ አረደበት /አረድ-ኧ-ብ-ኧት/(slaughters) ፣ ላከለት /ላክ-ኧ-ል-ኧት/(sends)", the sentence is called simple sentence. The morphemes (/-ኧች/፣ /-ኧ/፣ /-ው/፣ /-ብ /እና /-ል/) added on the verbs give us information about the subject, the modifier and object of the sentence. In the first sentence /-ኧች/, in the second and the third sentence /-ኧ/ are reflexives which indicate the subjects of the sentence. /-ው/ that is in the first sentence indicate reflexive of the object. /-ብ / and /-ል/ in the second and the third sentences are reflexive of the modifier's propositional phrase [18].

Therefore, in order to know whether the noun phrase is subject or object and the propositional phrase is modifier in a sentence, first we must show the position in the parse tree then we must show the morphological reflection. We can call the two features structural and morphological, the best one is morphological criterion because all noun phrases that came at start of a sentence may not be subject [18]

Simple sentence is classified into four kinds called declarative sentence, interrogative sentence, imperative sentence and negative sentence [18, 37].

Declarative Sentence

A simple sentence can explain about a subject and the explanation may be about the state of beingness, change or assimilation such sentences are called declarative sentences [37]. For example:

- ካሳ ተማሪ ነው።(Kassa is a student)

- ካሳ ጎበዝ ነው።(Kassa is clever)
- ካሳ ወታደር ሆነ።(Kassa becomes a soldier)
- አስቴር እናቷን ትመስላለች።(Aster looks like her mother)

Examples 1 and 2 explain about what Kasa is (the state of beingness). When we say Kasa becomes a soldier in example 3, we want to show the change from not being a soldier to becoming a soldier. Example 4 shows similarity between Aster and her mother.

In Amharic language grammatical voice of the verb can be active, passive or middle. They are defined by how they are used in relation to the subject.

Active Voice

In active voice, subject of the sentence is the doer of an action. Example 1:

- ተማሪው ደብተሩን ቀደደው። (The student tears his exercise book)
- አስተማሪው ለተማሪዎቹ መፅሃፍ ሰጣቸው። (The teacher gives a book for his students)

The examples clearly indicate as the subjects of the sentences perform the action expressed in the verb on others. For example, in the sentence "ተማሪው ደብተሩን ቀደደው" (The student tears his exercise book) the action of tearing is done by "ተማሪው" (the student) the receiver of the action "ደብተሩ" (his exercise book). As a result the subject "ተማሪው" (the student) is doer of the action and the object "ደብተሩ" (his exercise book) is receiver of the action.

In the second example, the action of giving is done by the subject "አስተማሪው" (the teacher) on the object "መፅሃፍ" (book). But, there are also receivers of the book. In this kind of sentence 3 main parts (subject, direct object and indirect object) are mandatory and the noun phrase which receives an action "መፅሃፍ" (book) becomes direct object and the prepositional phrase "ለተማሪዎቹ"(for his students) become indirect object. Example 2:

- ካሳ ወደ ገበያ ሄደ።(Kasa goes to market)
- ልጆቹ ዛፉ ላይ ወጡ። (The children climb on the tree)

The first example indicates as the subject of the sentence goes from one place to the other and the action of going is shown by the verb “ሄደ” (goes). These kinds of verbs need complement (prepositional phrase). So, the prepositional phrases “ወደ ገበያ and ዛፉ ላይ” (to market and on the tree) are used in the given examples. For example, the verb “ሄደ” (goes) is joined with prepositional phrase “ወደ ገበያ” (to market) and form the sub verbal phrase. Then it is connected with the subject “ካሳ” (kassa) to form the sentence “ካሳ ወደ ገበያ ሄደ” (Kassa goes to market) and answers the question “Where does kassa go?” Example 3:

- ልጆቹ በሩ ላይ ቆሙ። (The children stand at the door)
- አስቴር አልጋው ላይ ወደቀች። (Aster falls on the bed)

A sentence like “ልጆቹ በሩ ላይ ቆሙ” (The children stand at the door) can show where the subject performs a specific activity and the place is indicated by the prepositional phrase used in the sentence. For example, in the above sentence the children perform the action of standing at the door, so “በሩ ላይ” (at the door) shows where the subject of the sentence performs the activity. Therefore the sentence that talks about the doer of an action is called active sentence [18].

Passive Voice

In passive voice a noun phrase that receives an action can be used as a subject of a sentence and the verb starts with a morpheme called / -ተ/. For example:

- አስቴር ተሹማለች።
- ቡናው ተወቅጧል።

In the examples, the subjects “አስቴር and ቡናው” are the receivers of the action not the doers. The doer is not clearly indicated. In addition, the verbs used in the sentences begin with the morpheme /-ተ /.

However, in the sentence “ካሳ አስቴርን ሾሟታል” the subject “ካሳ” is a doer and the object “አስቴር” is a receiver. Then, we can understand that the subject of a sentence in passive voice can be the object of a sentence in active.

Middle Voice

Middle voice is in the middle between the active and the passive voices because the subject often cannot be categorized as either agent or patient but may have elements of both.

Interrogative Sentence

Interrogative sentence can be used to ask a question for knowing something new or for checking what is known before. The question can be about the subject, the complement or the verb. Interrogative sentence ends with a question mark. For example:

- ካሳ መቼ መጣ ? (When does Kassa come?)
- ካሳ መጣ ? (Does Kassa come)

According to the first question the requester doesn’t know the time when kasa comes?

Negative Sentences (አሉታዊ አረፍተ ነገር)

These are sentences that make declarative sentences negative in meaning. They simply negate a declarative sentence. For example, the sentence “ካሳ ወደ ትምህርት ቤት አልሄደም” (Kassa didn’t go to school) is negative because the prefix “አል” in the verb “አል-ሄደም” makes the sentence negative declarative sentence.

Complex Sentences

Complex sentences are formed from either complex noun phrase or complex verb phrase or both. In other words, a complex sentence can have a complex NP and a simple VP, a simple NP and a complex VP or both complex NP and complex VP. Complex NPs contain at least one embedded sentence which can be complement or other type phrase. On the other hand, complex VPs contain at least one sentence or more than one verb. Let us look at the following examples which

are taken from [37] to see the formation of complex sentences from noun phrases and verb phrases.

Example 1: “ካሳ የገባበት የሳር ቤት በጣም ትልቅ ነው” (the thatched house that Kassa has entered is so big). In this sentence the head of the noun phrase is “ካሳ የገባበት የሳር ቤት” (the thatched house that Kassa has entered). The head with the complement “የሳር” (thatched) forms simple noun phrase “የሳር ቤት” (thatched house) and this noun phrase is combined with the embedded sentence or clause “ካሳ የገባበት” (that Kassa has entered) to form complex noun phrase. But the clause that makes the complex phrase is dependent which is identified by the morpheme “የ” (that).

CHAPTER THREE: RELATED WORK

3.1 Introduction

There has been growing interest in domain independent semantic analysis, fed off recent efforts given to semantic role labeling because semantic role labeling is believed to be an important task on the road to natural language understanding, and has immediate applications in tasks such as information extraction and question answering [38]. Although many rule-based techniques like LinkParser, MiniPar and Lexical Functional Grammar have been traditionally used for this task, success has also been achieved by statistical techniques [2]. Therefore, in this chapter, we review research works which have been done on the area of semantic role labeling using different classification methods.

3.2 Semantic Roles Labeling Using Probability Estimation

The foundations for automatic semantic role labeling for FrameNet were laid in 2002 by Gildea and Jurafsky [10].They were pioneer in the semantic role labeling task. Their system was based on statistical classifiers that were capable of assigning FrameNet roles to syntactically parsed sentences automatically. It is trained on roughly 50,000 sentences that were hand-annotated with semantic roles by the FrameNet semantic labeling project. The classifier was trained on a labeled training set and tested on a held-out portion of the data. The following procedures were employed in the study. First, the sentence is parsed by a constituent parser to obtain a parse tree. Then Phrase Type, Governing Category, Parse Tree Path, Position, Voice and Head Word features were derived. Posterior probability distributions were used for the classification task. Probability model considers the question of finding the boundaries of frame elements separately from the question of finding the correct label for a frame element, although similar features were used for both tasks. In order to automatically label the semantic role of a constituent, a probability distribution indicating how likely the constituent is to fill each possible role is estimated, given the features and the predicate, or target word, *t*: P (*r* |*h; pt, gov, position, voice, t*). It would be possible to calculate this distribution directly from the training data by counting the number of times each role appears with a combination of features, and dividing by the total number of times the combination of features appears:

$$p(r|h;pt,gov,potion,voice,t) = \#(r,h,pt,gov,postion,voice,t/\#(h,pt,gov,postion,voice,t) \quad (4)$$

However, the data was scarsed for estimating the conditional probability of a label given all the above features together. This was because, a particular combination of the above six features along with the target word occurs rarely in the dataset. To overcome this, conditional probabilities of the frame element labels were given subsets of the above features like $p(role|targetword)$, $p(role|path,targetword)$, etc. were computed. These were combined using different strategies like equal linear interpolation, EM linear interpolation, Geometric Mean, Back-off linear interpolation and Back-off geometric mean. This strategy gives a significant improvement in performance over the baseline approach of directly estimating the conditional probability of the labels given all the six features in the conditioning set. In spite of the pioneering nature of their research, the authors achieved quite impressive results: 82% accuracy in identifying the semantic role of pre-segmented constituents, 65% precision and 61% recall at the more difficult task of simultaneously segmenting constituents and identifying their semantic role.

Wang [39] proposed semantic role labeling in Chinese using HowNet. The author demonstrated the benefit of applying the knowledge from HowNet to Semantic Role Labeling by experimenting with four selected Chinese words. The researcher used a statistical approach and extracted various lexical and syntactic features, including the phrase type of each constituent, the headword, and the position and distance from the predicate to the constituent in question and voice. The study also revealed that the system can be improved by applying more information from HowNet, introducing full parsing information, enriching the feature set, and using more appropriate probability estimation model. A set of 11 predicate-independent abstract semantic roles such as Agent, Patient, Theme, Experiencer, Instrument, Location, Source, Goal, Time, Frequency and Quantity used in the study. To develop training set four Chinese verbs were selected by considering three factors (1) Frequency, the verbs should be frequent enough to provide sufficient training data, (2) Syntactic diversity, to select verbs with different numbers of arguments, and with various argument patterns, which were representative of the variety of syntactic behavior in the language, (3) Word sense, verbs that varied in their number of word senses were preferred. After selecting the verbs, all sentences containing these four verbs were

selected from a corpus. Totally, they had 259 sentences then they splited the data for each verb into two parts: 80% for training and 20% for testing. To identify the arguments a number of features were extracted from the input sentence and parsed. Phrase type, position and distance, voice, headword, and backoff models for headword features and HowNet semantic resource were used in the study. The probability of the event that the constituent was to fill each possible role to label the semantic role of a constituent estimated automatically. If the features and the predicate, or target word, *t* is given probability calculated as follow:

$$p(r \mid h, pt, p\&d, voice, t) = f(r, h, pt, p\&d, voice, t / f(h, pt, p\&d, voice, t) \quad (5)$$

Where ***r*** indicates semantic role, ***h*** for headword and **h*&d*** for position and distance. The probability would be estimated from the training set by counting the number of times each role appears with a combination of features and dividing by the total number of times the combination of features appears. It was difficult to implement in practice, however, because of data sparseness. So, the researcher approximates the probability to the product of four independent probabilities and smoothed them by giving a very small number [39].

$$p(r|h,t) * p(r|pt,t) * p(r|pd,t) * p(r|v,t) \quad (6)$$

The system achieved a high precision and recall for some target words because of its strong patterns in predicate-argument structure. An average of 78.5% precision and 68.9% recall were achieved.

3.3 Semantic Role Labeling Using Support Vector Machines

Pradhan *et al* [11] proposed a machine learning algorithm for shallow semantic parsing by extending the work of Gildea and Jurafsky [10]. Their algorithm was based on Support Vector Machines, which show an improvement in performance over earlier classifiers. Performance improvements were achieved through a number of new features and measure the ability to generalize to a new test set drawn from the AQUAINT corpus. PropBank corpus was used in the study. The authors viewed problem of shallow semantic parsing as three different tasks:

- **Argument Identification**: Identify parsed constituents in the sentence that represent semantic arguments of a given predicate.
- **Argument Classification**: Given constituents known to represent arguments of a predicate, assign the appropriate argument labels to them.
- **Argument Identification and Classification**: A combination of the above two tasks.

Each node in the parse tree can be classified as either one that represents a semantic argument (i.e., a NON-NULL node) or one that does not represent any semantic argument (i.e., a NULL node). The NON-NULL nodes can then be further classified into the set of argument labels. The parsing problem was formulated as a multi-class classification problem and Support Vector Machine (SVMs) classifier was used. Since SVMs are binary classifiers, the multiclass problem is converted into a number of binary-class problems. The ONE *vs* ALL (OVA) formalism was used, which involves training *n* binary classifiers for *n*-class problem. In addition to feature described by Gildea and Jurafsky [10], several new features were tested in the study and which are listed below. From these features, two of them (named entities in constituents and head word part of speech) were obtained from the literature. Others were novel features of the author.

- **Named Entities in Constituents:** Named entities (PERSON, ORGANIZATION, LOCATION,PERCENT, MONEY, TIME, DATE)
- **Head Word POS**: Part of Speech tag of the headword.
- **Verb Clustering**: Predicate clustering to counter unknown predicates.
- **Partial path:** To avoid data sparsity parse tree paths in partial forms.
- **Verb Sense:** Information Word sense of the predicate.
- **Head word of prepositional phrases**: Replacing the preposition with the first noun as the head word.
- **First and Last word/POS in constituent**: Found to be discriminative.
- **Ordinal constituent position**: To avoid false positives of elements far away from predicate.
- **Constituent Tree Distance:** Finer way of depicting an existing feature.
- **Constituent relative features:** Features of parents and siblings.
- **Temporal cue words:** Temporal words not captured by NER.

- **Dynamic class context**: Hypotheses of at most two previous nodes belonging to the same tree.

The authors evaluated the results using both hand-corrected TreeBank syntactic parser, and actual parser from the Charniak parser. The Baseline system achieved 90.9% precision, 89.8% recall and 90.4% F-score in argument identification, 87.9% accuracy in argument classification and 83.3% precision, 78.5% recall and 80.8% F-score in the combination of both tasks for all ARGs using hand-corrected parser. Their system also achieved 94.7% precision, 90.1% recall and 92.3% F-score in argument identification, 91.4% accuracy in argument classification and 88.4% precision, 84.1% recall and 86.2% F-score in the combination of both task for CORE ARGs using hand-corrected parser. Performance decreased when they were using automatically generated parser as 89.3% precision, 82.9% recall and 86.0% F-score in argument identification, 90.0% accuracy in argument classification and 84.0% precision, 75.3% recall and 79.4% F-score in the combination of both task for all ARGs using automatic parser. 92.0% precision, 83.3% recall and 87.4% F-score in argument identification, 90.5% accuracy in argument classification and 86.4% precision, 78.4% recall and 82.2% F-score in the combination of both tasks were achieved for CORE ARGs using automatic parser.

Sun and Jurafsky [12] presented the preliminary work on Chinese SRL without any large semantically annotated corpus of Chinese. In this study, the questions of assigning semantic roles to Chinese sentences were addressed. Good semantic parsing results for Chinese can be achieved with a small 1100 sentence training set. In order to extract features from Chinese, porting the Collins parser to Chinese was defined, resulting in the best performance currently reported on Chinese syntactic parsing. The SVM algorithm proposed for English by Pradhan *et al* [11] was used in their study. First, a small 1100-sentence Chinese corpus was labeled according to principles from the English and (in-progress) Chinese PropBanks. Second, the features used by SVMs classifier were introduced and show the performance on semantic parsing for both seen and unseen verbs for given hand-corrected (Chinese TreeBank) syntactic parser. Then port of the Collins parser is labeled to Chinese. Finally, SVM semantic parser is applied to a matching English corpus, and discusses the differences between English and Chinese that lead to significantly better performance on Chinese. The authors investigated different features that were proposed for English in Chinese; some acted quite similarly to English, while others

showed interesting differences. Features that acted similarly to English include the target verb, the phrase type, the syntactic category of the constituent (NP, PP, etc.), and the sub categorization of the target verb. Five features (path, position, governing category, head word, and voice) showed interesting patterns. In Chinese the marked passive voice was indicated by the use of the preposition "被/by". This passive, however, is seldom used in Chinese text because of this voice feature didn't use in the study. Their system achieved 69.8% precision, 50.7% recall and 58.3% F-score for unseen verbs. This result was based on the use of perfect (hand-corrected) parser drawn from the Penn Chinese Treebank.

3.4 Semantic Role Labeling using Memory Based Learning

Ghalibaf *et al* [40] proposed the first Persian corpus based SRL system using memory-based learning model and standard features. The input data was drawn from Hamshahri corpus. It was hand-labeled with required syntactic and semantic information. Twelve semantic roles were proposed in the study which were divided into two classes: primary and general roles.

- **Primary Roles**: are predicate specific such as: Agent, Patient, Source, Goal, Topic, Percept, Instrument and Beneficiary.
- **General Roles**: are assumed to apply across all verbs. They were optional for an event but supplied more information about an event including: Location, Time, Manner and Reason.

The authors implemented SRL in two phase architecture. First, the arguments were identified using a shallow syntactic parser or chunker and then labeled them with appropriate semantic role with respect to the predicate of the sentence using a machine learning method to distinguish different roles such as Agent, Goal, etc. Both phases were treated as a multi-class classification problem. The classifier was trained in a supervised manner from human-annotated data and using memory-based learning. Some syntactic properties of arguments were used as feature set. The authors stated that, the feature set plays an important role in MBL performance and mainly they selected features from the review of previous literature, investigated each of these features in Persian, some acted quite similarly to English, while others showed interesting differences. Current argument phrase type **(**the syntactic type of constituents (NP,PP,VP,ADV,SP), Previous argument phrase type, Next argument phrase type, Position and Verb Class (These classes are

based on the semantic roles each verb can take) features showed interesting patterns. The experiments were carried out with the TiMBL software. Regarding the learning algorithm, the IB1 classifier was used and parameterized by using overlap as the similarity metric, information gain for feature weighting, using 1 k-nearest neighbors, and weighting the class vote of neighbors as a function of their inverse linear distance. They used three measures for the evaluation of their system: precision, recall and F-Score (a combined measure). The results showed an F-score of 90.3% on argument boundary detection task and an F-score of 87.4% on semantic role labeling task using Gold-standard parses. An overall system performance showed an F-score of 83.8% on complete semantic role labeling system, i.e., boundary plus classification.

Stevens [5] presented an approach to automatic semantic role labeling in Dutch corpora. The goal was to annotate a small portion of the D-Coi corpus with semantic roles automatically by using TiMBL, a memory based classifier. Since there was no semantically annotated Dutch corpus available that can be used as training data, a novel approach to rule-based tagging based on Alpino dependency trees was proposed. Phrase type (category label), dependency label, POS-tag, position, argument head-word, Head-word POS tag and category label and dependency label in combined features were used to describe the candidate argument. For the task of automatic semantic role labeling, MBL classifier was employed. The evaluation result showed that the TiMBL performance on training data using the LOO method was 65.05% precision, 65.53% recall and 65.29% F-score. The best performing system reached F-score of 80%. The system would rank in the lower regions, this is due to the following reasons:

- The candidate argument selection method used was not perfect. The author did not perform any pre-post-processing steps.
- The author did not use a very basic set of features.
- The author did not attempt to optimize TiMBL parameters.
- The size of training set was small.

3.5 Semantic Role Labeling Using Artificial Neural Network

Wang [41] proposed a new method for labeling Chinese with semantic roles neither using syntactic parsing nor part of speech tagging technologies. The task was divided into two

subtasks, clustering and labeling. Clustering was aimed at partially replacing syntactic parsing, during which similar sentences are clustered together. In the labeling step, artificial neural networks was planted as many as the number of clusters, each of which takes charge of summing up features of chunks of a sentence and then labeling them with semantic roles. The researcher hypothesizes that particles and verb meaning were closely related to semantic roles arguments. Particles and verbs' meaning were the most important features used in clustering and in identifying semantic roles. In the experiment, more than one thousand Chinese clausal sentences are annotated manually, in which assertions, imperatives, queries are all included. All the syntactic elements are sorted into four categories: argument, predicate verb, particle, and marker. Besides, a mini-thesaurus of 600 Chinese common verbs were compiled, which were used as the principal reference during annotating. The experiment result shows this method is useful; and 83.8% correctness on average was achieved. This achievement mainly comes from the effectiveness of clustering. Clustering makes the mess of sentences display orderly so that some useful features can be summed up. Moreover, some irregular sentences with lower probability features will be easily detected; so it can be well labeled by artificial neural network either. However, the system performs unsteadily when the size of clusters is taken into account. It was probably because shorter sentences tend to be clustered easily that noise in the clusters hindered artificial neural networks to learn useful knowledge.

3.6 Summary

As we have discussed in this chapter, sematic role labeler is developed in different approaches for different languages around the world. In Table 3.1 we have tried to summarize some of the works which are related to our work.

Table 3.1: Related Work Summary

Language	Approach	Feature used	Performance for semantic role classification
English	Probability Estimation	-Phrase Type - Governing Category - Parse Tree Path - Position - Voice - Head Word	82% accuracy, 65% precision and 61% recall
Chinese	Probability Estimation	- Phrase type -Headword - Position & distance - Voice	78.5% precision and 68.9% recall
English	SVMs	-Feature described by Gildea and Jurafsky - Named Entities in Constituents - Head Word POS - Verb Clustering - Partial path - Verb Sense - Head word of prepositional phrase - First and Last word/POS in constituent - Ordinal constituent position - Constituent Tree Distance	87.9% accuracy
Chinese	SVMs	- path -position -head word	69.8% Precision 50.7% Recall

		- governing category - voice	and 58.3% F-score
Persian	MBL	- Current phrase type - Previous argument phrase type - Next argument phrase type - Position - Verb Class	87.4% F-score
Dutch	MBL	- Phrase type - Dependency label - POS-tag - Position- Argument head-word - Head-word POS tag - category label and dependency label	65.05% precision, 65.53% recall And 65.29% F-score
Chinese	ANN	- Position - Verb meaning - Markers<0>.	83.8% accuracy

Although a number of related works have been conducted on the area of semantic role labeling for other languages, these could not be directly applied to Amharic language due to the difference in the general role of language specific characteristics in the extraction of semantic content and morphological complexity of Amharic language. Hence, this study will fill the gap and address the issue of semantic role labeling for Amharic text.

CHAPTER FOUR: DESIGN OF SEMATIC ROLE LABELER FOR AMHARIC TEXT

4.1 Introduction

In this chapter, we discuss about the overall design of the proposed semantic role labeler for Amharic text. First, we illustrate the generalized architecture of the proposed system, and then we describe the three main component of the proposed system such us the text pre-processing, the training and the semantic role labeling components along with their sub-components.

4.2 Architecture of the System

Figure 4.1 illustrates the generalized architecture of the semantic role labeler for Amharic text using memory based learning. It consists of three components text pre-processing, training and semantic role labeling. The text pre-processing component accepts the input text, POS tagged each word in text with appropriate POS tag and parses the text with phrase structure schema in which an X-bar theory is used to flesh out phrase structure. The second component is training. In the training component, the system was trained on the text where segments of texts are already correctly labeled (with semantic roles in this task). It reads the text training input and collects the knowledge about the occurrences of labels. The third component is semantic role labeling. It reads a new text for which the labels are to be automatically assigned and attempts to predict the correct label for every given segment of the text using the information available in the text and the knowledge acquired in training.

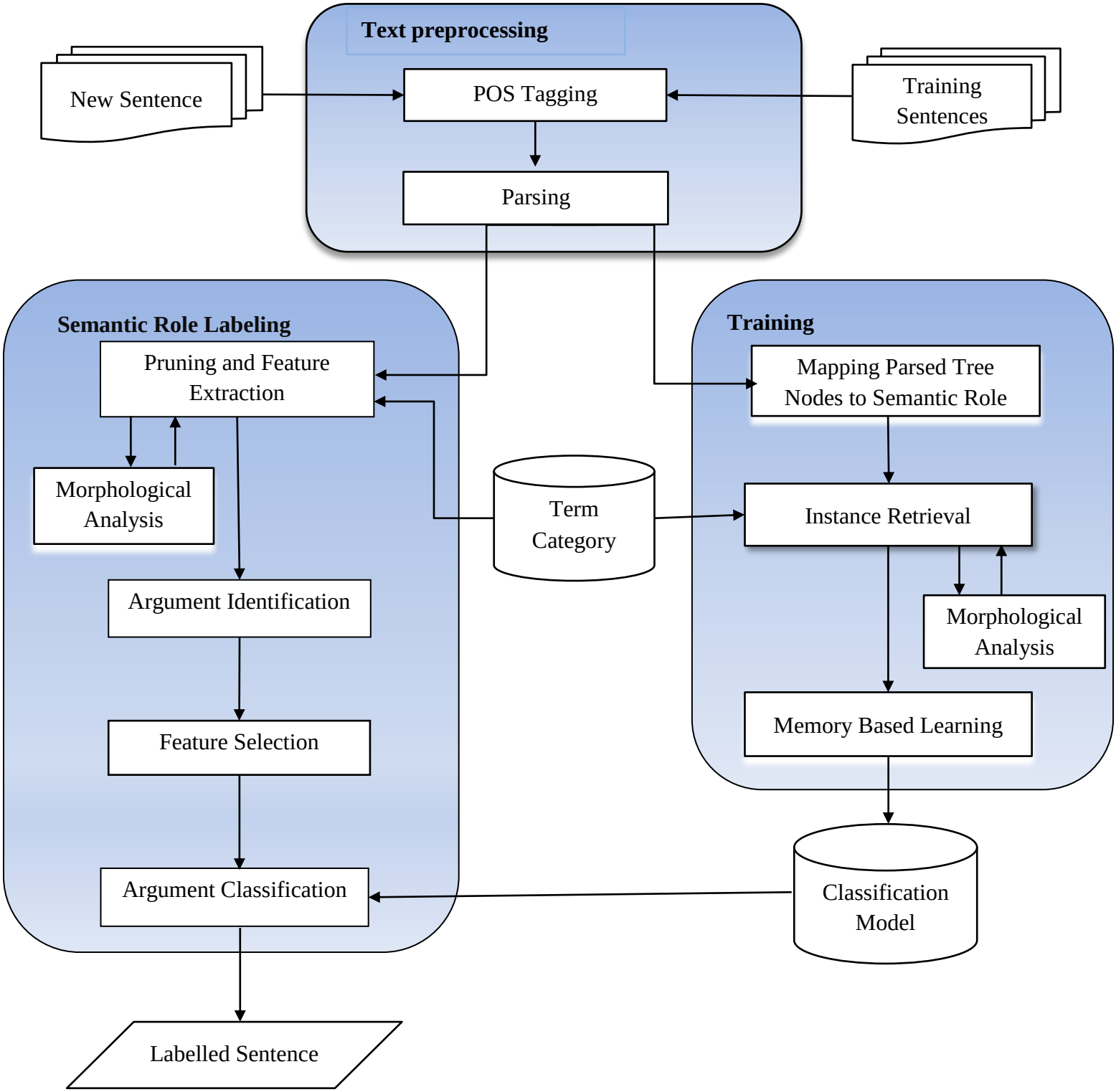

Figure 4.1: General Architecture of the Proposed System

4.3 Text Preprocessing

The input text for both the training and semantic role labeling pass through text preprocessing. The following two tasks are performed in order to make ready the input text for further processing.

POS Tagging: It is performed to assign each input sentence words with their part-of-speech. It helps us to correctly classify the semantic role of the sentence constituents. A list of POS tag used in our system and their description is shown in Appendix A.

Parsing: We employed parsing to divide a sentence into segments which correspond to certain syntactic units (noun, verb, preposition phrase, etc.). These segments represent semantic arguments of a given predicate (often shown by the verb). As we have discussed in Section 2.4.3 existing parsers can be divided in to two types according to their annotation schema phrase-structure and dependency structure. Dependency structures are more abstract and rich in information on the relation between words and provide a good starting point for semantic annotation. However, due to lack of getting enough work that was done on Amharic dependency structures and the nature of phrase-structure is easily converted in to dependency structures using simple algorithm, our work is based on first type phrase-structure, in order to retrieve possible constituent (phrase) from the sentence.

Figure 4.2 shows phrase structure for simple Amharic sentence “ካሳ ምሳውን በላ / Kassa məsawənə bäla/ Kassa eats his lunch”.

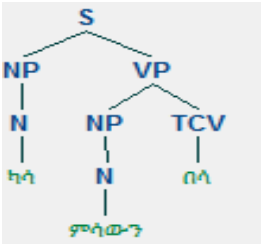

Figure 4.2: Phrase Structure for an Amharic Sentence

The phrase structure illustrated in Figure 4.2 can be represented in bracketed string format as follows: (S(NP(N ካሳ))(VP(NP(N ምሳውን))(TCV በላ))

The quality of the parser has a major impact on the final performance of SRL system, and the percentage of correct constituents that is generated by the syntactic parser also defines the recall upper bound of SRL system. Therefore, for the good quality SRL the text preprocessing task was done manually.

4.4 Training

4.4.1 Mapping Parsed Tree Nodes to Semantic Role

In this section, we try to make the relation between parsed tree node and semantic role. PropBank annotation method is select for the study. The head node of a sentence is a verb. In general, verbs correspond to the predicates in PropBank propositions. Table 4.1 show Basic Relation between Parse tree Node and PropBank Roles. The argument structure of a verb is encoded by its complements and modifiers. Complements are verbal arguments, therefore correspond to numbered arguments in PropBank. Modifiers in parse tree have the same function as modifiers in PropBank propositions and correspond to ARGMs.

Table 4.1: Basic Relation between Parse tree Node and PropBank Roles

PropBank Label	Phrase Structure Node
ARG0 . . . ARGn	Complement
ARGM-xxx	Modifier
Predicate	Head

The relation between subject/object complements and PropBank arguments is illustrated in Figure 4.3.

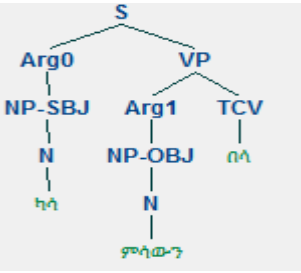

Figure 4.3: Parse Tree with PropBank Role Layer

As a result of this process, we get a parse tree with PropBank semantic role layer. Mapping was conducted manually. Figure 4.4 illustrates semantic role annotated sentence in string format which shown in Figure 4.3 and Appendix C.

Figure 4.4: Parse Tree with PropBank Role Layer in String Format

4.4.2 Instance Retrieval

After the sentences are annotated with semantic role, features are extracted automatically from the manually annotated sentence to make instances using an Algorithm 4.4. These features, capture the basic information about each input that should be used to classify it. The features we used represent various aspects of the syntactic structure of the sentence as well as lexical information. The syntactic information can serve as an indicator for the semantic role.

In Section 2.7, we have seen different features through reviewing previous works about SRL. We investigated some of the features in Amharic. The six features that have good discriminative power for Amharic were selected for this study. These selected features for the study were clustered into three. The first cluster characterizes target predicate, the second cluster characterizes the candidate argument and the last cluster characterizes relationship between predicate and candidate argument.

i. Features that characterize target predicate

Verb Class: These classes were based on the semantic roles each verb can take. Although several classifications are now available for other language verbs such us for Spanish and English, there is no such classification for Amharic verbs. In this work, we provided a classification for Amharic verbs that consists of 14 classes see Appendix B. Which are grouped on the basis of both syntactic and semantic alternations. For this purpose, we first grouped a number of Amharic verbs according to the number of syntactic arguments and then classified them into smaller groups which have similar set of semantic roles. Having a membership in a particular class says something about the predicate-argument structure of a verb. The class

information tells the system how to label the semantic roles of the verbs belonging to a particular class.

For example, verb ‘መደበ/mädäbä (assign) belongs to verb class of ditransitive-agentive-benefactive (DABV) which is described as follows:

Verb class DABV: [+Agent, +Theme, +Beneficiary]

Semantic roles for the sentence ‘ የትምህርት ሚኒስቴር ለኤች አይ ቪ ኤድስ መከላከያ 10.8 ሚሊዮን ብር መደበ / yätəməhərətə minisəterə läʾečə ʾäyə vī ʾedəsə mäkälakäya 10.8 miliyänə bərə mädäbä (The Ministry of Education assigned 10.8 million birr for HIV-AIDS Prevention) are as follows:

Argument	**Semantic Role**
የትምህርት ሚኒስቴር	Agent
ለኤች አይ ቪ ኤድስ መከላከያ	Beneficiary
10.8 ሚሊዮን ብር	Theme
መደበ	Predicate

The example demonstrates the fact that arguments don't necessarily appear in the order that they are written in the role set. The complete list of verb class and their semantic classes are given in Appendix B.

Voice of the Verb: It refers to grammatical voice of the predicate, active, passive, middle. The active and passive verb forms in Amharic share the same predicate argument structure; but the grammatical functions may be mapped to different sets of semantic roles. Algorithm 4.1 illustrates how grammatical voice of the verb done for this study. In active voice the subject of the sentence is the doer of an action, in passive voice a noun phrase that receives an action can be used as a subject of a sentence and the verb starts with a morpheme called / -ተ/. The middle voice is said to be in the middle between the active and the passive voices because the subject often cannot be categorized as either agent or patient but may have elements of both. Detail description about grammatical voice of verb in Amharic is given in section 2.9. In order to identify grammatical voice of the verb “*HornMorpho*” morphological analyzer [51] is used.

```
        INPUT: Parse Tree, Non-Passive list, HornMorpho
        OUTPUT: Voice of the verb
From given tree extract verb and its POS
If Verb POS = = USV or UMV or AUV or EUV // class of the verb see Appendex B
   Voice="middle"
Else
   Check voice from  HornMorpho
     If it return "smp"
       Voice = = "active"
      Else
        Check it in non_passive verb list
         If it exist  in a list
              Voice = = "active"
         Else
              Voice = ="passive"
```

Algorithm 4.1: Grammatical Voice of Verb Identification

ii. Features that characterize the candidate argument

Phrase Type: Syntactic category of the constituent such us NP, PP, ADJP, etc. It is extracted easily from a parse tree for a given candidate argument by simply find the candidate argument label using pre-defined ***argument.lable()*** *method*. The argument label in a parse tree is phrase type of the argument.

Head Word Term Category: head word of phrase is used as a feature. However, instead of simply using the head word itself, we give the same term category for similar words. For example, we give Person (P) term category for the name of persons. However, if the phrase type of the candidate argument is PP, the head word itself is used as a feature.

The head of a phrase is the word that determines the syntactic category of the phrase. For example, the head of the noun phrase "የ ወርቅ ቀለበት" is the noun "ቀለበት". In Amharic Head of a phrase is usually the last word of the phrase. But it may not always be true for prepositional phrase. Prepositional phrase head may be at the beginning. Many theories of syntax represent heads by means of tree structures. Thus, the head word is easily extracted from parse tree using Algorithm 4.2. After the head word is identified term category of the head word is searched. In order to find it first head word is set to "H*ornMorpho*" it return the stem word, second check the

stem word term category in “Term Category” data base. If the term category of the stem word exist it is taken else stem of head word itself taken.

```
    INPUT: parse tree, candidate argument
    OUTPUT: head word
If the phrase type is PP
        For each level on parse tree
                For node on current level
                        If current node is P
                                If parent node is PP
                                        Return Child of P
                                End If
                        End If
                End For
        End For
End If
Else
Return right most leaf node
```

Algorithm 4.2: Head Word Identification

Prepositional Phrase Complement: When a verb’s argument is a prepositional phrase (PP), the PP’s head word is the preposition. While this can often be a reliable indicator of semantic role (for example “ለ/lä” usually indicate beneficiary), most prepositions can be used in many different ways, and the sense can be determined by the preposition’s complement. For example, “በ 11 ሰአት/bä 11 säʾätə” indicates time, while በ አዲስ አበባ/ bä ʾädisə ʾäbäba” indicates location. Once the head word of the phrase is identified finding complement from parse tree is easier. There for in order to find complement of PP first the head word of phrase identified using Algorithm 4.2 then find the sister of head word sister of head word is the same as complement of the phrase.

iii. Features that capture the relationship between predicate and candidate argument

Syntactic function: Syntactic function of candidate argument in the sentence (subject, object, direct object, indirect object, modifier or complement)

```
INPUT: Parse Tree , Argument list , Argument Phrase type, Verb POS
OUTPUT: Syntactic Function
From parse tree extract sibling of Argument list
For each argument i in Argument list
    If  the sibling of i = VP
        Find the parent of i
        If the parent of i = S
            Syntactic function of i = Subject
        Else If the parent of i  = VP
            Syntactic function of i = specified
        Else
            Syntactic function of candidate argument = modifier
    Else If sibling of i= TCV or TAV
        Syntactic function of i =  object
    Else If sibling of i= DABV or DALV
        Phrase type i=get Phrase_type(i)
        If phrase type of i = NP //noun phrase
            Syntactic function of i = direct object
        Else if  phrase type of i = PP (prepositional phrase)
            Syntactic function of i = indirect object
    Else
        Syntactic function of i = complement
End for
```

Algorithm 4.3: Syntactic Function of Argument Identification

Automatic Feature Extraction Process

The learning tool used in our work is TiMBL (Tilburg Memory Based Learner). To be able to train a TiMBL classifier, a file with training data is needed. Training data is represented as a text file containing instances. Each line in the text file represents a single instance. An instance consists of a set of features separated by commas and a target class. Once the given sentence is stored and it is clear how instances will be encoded, such an instance can be extracted from the annotated corpus automatically. For example, the following instances can be extracted from the tree in Figure 4.3.

TCV, Active, P, NP, #, SUBJ, ARG0-AGT

TCV, Active, FO, NP, #, OBJ, ARG1-PAT

This output is suitable input to the TiMBL classification tool. Let's have a closer look at the values of first instance (from left to right):

- **TCV**: Verb POS "በላ/bäla" (eat) it is transitive causative verb class
- **Active:** Grammatical voice of the verb "በላ/bäla" (eat)
- **P**: Term category of the constituent "ካሳ /Kasa" it is in person name category (P)
- **NP:** Phrase type of the constituent "ካሳ /Kasa" (his lunch)
- #: Null because the argument node ("ካሳ /Kasa" (his lunch) is not PP
- **SUBJ**: Syntactic function of the argument "ካሳ /Kasa" in the sentence is subject
- **ARG0-AGT**: The target class

The extraction of instance from the annotated corpus can be done automatically. In order to do this, instance retriever Algorithm 4.4 is used. It is implemented using Python programing language, which is able to convert an annotated corpus into a C4.5 instance base.

Input: Parse tree with PropBank role layer
Output: Instance in C4.5 format

```
Current_node=get_verb(parse_tree)
Current_node_voice=get voice (Current_node) //using Algorithm 4.1
Current_node_POS=get POS(Current_node)
Current_node_sister=sisters_collector(current_node)
For each sister S current_node sisters
        S_POS=get POS(Sister node)// S POS is class of child node
        S child=get child (S)
        child phrase_type=get phrase_type(S child)
        child syntactic function= get syntactic function (s child) //using Algorithm 4.3
        child head word =get head word(child)// using Algorithm 4.2
End for
Current_node_parent =get_parent(current_node)
If current node parent !='S'
        Curremt_node=current_node_parent
        Goto line 4
End if
```

Algorithm 4.4: Instance Retrieval

4.4.3 Morphological Analysis

In our system, the morphological analyzer is used in two places; in the construction of the learning model and in semantic role labeling. In both phase the main purpose of the analyzer is to find the grammatical voice of the given verb. Unlike other language such as English grammatical voice of Amharic sentence doesn't identified without the use of morphological analyzer. Hence in our study we used the morphological analyzer known as HornMorpho developed by Gasser [48]. Algorithm 4.1 shows how the morphological analyzer was used in our system.

4.4.4 Memory Based Learning

We are choosing to use memory based learning algorithm (MBL), which use specific instances rather than pre-compiled abstractions during prediction tasks. This is arguably a useful property for NLP tasks. MBL algorithms take a set of examples (fixed-length patterns of feature-values and their associated class) as input, and produce a classifier that can classify new, previously unseen, input patterns. Our MBL system is based on the use of TiMBL. It implements several memory-based learning algorithms (**IB1**, IB2, IGTREE, TRIBL and TRIBL2).

The memory-based learning algorithm used here are IB1 illustrated in algorithm 4.1 adopted from [43]. It constructs a data base of instances in memory during learning. Training instances are represented by tree structures. The structure starts by creating a root node. From the root node all values of a chosen feature is grown as an arc ending in a new node. The arcs are labeled with the feature value. Information about the number of training instances leading up to the node and class-information is stored in the nodes. From the nodes on the first level of the tree, branches are grown to all the values of the second feature that occur in the training set in combination with the value of the previous node.

IB1 searches the tree made on the basis of the training instances exhaustively. It commits to matches between a test instance's feature value and a feature value in the tree. The most important feature is compared first. The gain ratio weight is a good measure for the importance of a feature in determining the classification of instances. When a mismatch occurs, the default classification of the mismatching node's mother is assigned to the test instance.

```
        Input: Training Set (TS)
        Output: Concept Description (CD)
CD ← 0
For each x ∈ Training Set do
            for each y ∈ CD do
            Sim[y] ← SimilarityCx, y)
            y_max ← some y ∈ CD with maximal Sim[y]
             if class(x) = class(y_max)
                then classification ← correct
                else classification ← incorrect
            CD ← CD U {x}
```

Algorithm 4.5: The IB1 Algorithm

The primary output of MBL algorithms is a *concept description (or concept)*. This is a function that maps instances to categories: given an instance drawn from the instance space, it yields a classification, which is the predicted value for this instance's category attribute.

An instance-based concept description includes a set of stored instances and, possibly, some information concerning their past performances during classification (e.g., their number of correct and incorrect classification predictions). This set of instances can change after each training instance is processed. However, MBL algorithms do not construct extensional concept descriptions. Instead, concept descriptions are determined by how the MBL algorithm's selected similarity and classification functions use the current set of saved instances. These functions are two of the three components in the following framework that describe all MBL algorithms:

Similarity Function: This computes the similarity between a training instance **i** and the instances in the concept description. Similarities are numeric-valued.
Classification Function: This receives the similarity function's results and the classification performance records of the instances in the concept description. It yields a classification for **i**.
Concept Description Updater: This maintains records on classification performance and decides which instances to include in the concept description. Inputs include **i**, the similarity results, the classification results, and a current concept description. It yields the modified concept description.

The IB1 Algorithm, described in Algorithm 4.1 is the simplest instance-based learning algorithm. In IB1 algorithm similarity calculated as follows it adopted from [43] where the instances are described by **n** attributes.

$$similarity(x, y) = -\sqrt{\sum_{i=1}^{n} f(x_i, y_i)} \quad (7)$$

Given two instances x and y, the similarity is computed using the function given above. i is the index to an attribute. For numeric attributes, use Euclidean distance. For non-numeric attributes is 1 if they are equal, 0 otherwise. For missing attribute values are assumed to be maximally different from the value present. If they are both missing, then f(xi, yi) yields 1. IB1 is identical to the nearest neighbor algorithm except that it normalizes its attributes' ranges, processes instances incrementally, and has a simple policy for tolerating missing values.

As a result of this process, we get a memory based learning model which will be used for semantic role labeling. Therefore, the implementation of the memory based SRL relies on this learning model. The output of the trained dataset is taken as a tree file to be used during semantic role labeling.

4.5 Semantic Role Labeling

The SRL detects the semantic arguments associated with the predicate or verb of a sentence and their classification into their specific roles. The training module is critical for SRL in order to successfully implement memory based SRL. Once the proper problem text pre-processing is performed, semantic role labeling is carried out. The semantic role labeling component has five modules: pruning and feature extraction, morphological analysis, argument identification, feature selection and argument classification.

4.5.1 Pruning and Feature Extraction

Once the new sentence is parsed, the next task will be pruning and feature extraction. In pruning and feature extraction module a sentence is given in parsed tree format all arguments related to the verb have to be identified along with features such as phrase type, grammatical functions, voice, head word and PP complement. For this study pruning heuristics by Xue and Palmer [24] is adapted, the heuristics first spot the verb and then extract the entire sister nodes along the verb

spine of the parse tree. We also expand the coverage by extracting all the immediate children of an ADVP, ADJP, PP and NP node along with a few additional syntactic features such as phrase type, grammatical functions, voice, head word and PP complement. Figure 4.5 illustrated how pruning works for simple Amharic sentences. Arguments related to the verb "ልኮላታል" is "ካሳ", "ሁለት ጊዜ", "በ ባንክ", "ለ አስቴር" and "ገንዘብ" . If the predicate is in active form, the first NP under S node is considered as the subject, otherwise, the PP with "በ" under VP node is considered as the real subject. NP under S node is marked as the real subject of the predicate "ልኮላታል" because the predicate "ልኮላታል" is in active form. The output of pruning for the sentence "ካሳ ሁለት ጊዜ በ ባንክ ለ አስቴር ገንዘብ ልኮላታል" is shown in figure 4.6.

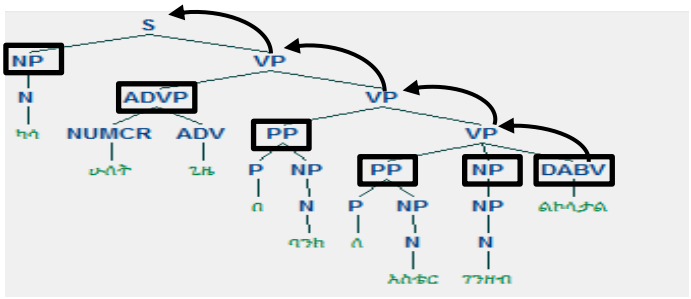

Figure 4.5 Pruning

***Input**: Parse Tree, predicate*
***Output**: Arguments and Arguments feature extracted from parse tree*
Current_node=get verb(parse tree)
Current node voice=get voice (Current_node) //using Algorithm 4.1
Current node POS=get POS(Current_node)
Line 4:
Current node sister=sisters collector(current_node)
For each sisters S current node sister
    S phrase type=get phrase type of (S)
    S syntactic function= get syntactic_function (S)// using Algorithm 4.3
    S head word =get head word(S)//using Algorithm 4.2
End for
Current node parent =get parent(current_node)
If current node parent !='S'
    Curremt_node=current node parent
    Goto line 4
End if

Algorithm 4.6: Pruning and Feature Extraction

4.5.2 Argument Identification

The argument identifier simply finds possible arguments for given predicates by browsing input parse-trees. These arguments represent semantic arguments of a given predicate (often shown by the verb). Argument identifier is used to filter out constituents that are clearly not semantic arguments to the predicate in question, it reduce the overall number of constituents to label and save training time.

Due to lack of available corpus to train the argument identifier we are not using ML approaches for argument identification. Therefore, in order to identify semantic role candidate: First syntactic function of candidate argument is searched in a parse tree. Then, ignore nodes with syntactic function specifier. Only consider complements and modifiers as possible arguments.

For example, consider the sentence "ካሳ ሁለት ጊዜ በ ባንክ ለአስቴር ገንዘብ ልኮላታል /Kasa hulätəgize bä banəkə läʾäsəterə gänəzäbə ləkolatalə". In this sentence arguments of the verb "ልኮላታል" are "ካሳ", "ሁለት ጊዜ", "በ ባንክ", "ለአስቴር" and "ገንዘብ" syntactic function of each argument based on Algorithm 4.3 are subject, specifier, modifier, indirect object and direct object respectively. Argument "ሁለት ጊዜ", is removed from the list because its syntactic function is specifier and others are considered as possible semantic role candidate.
For this work syntactic function for a given argument find from a parse tree using Algorithm 4.3. Then to get the possible list of semantic role candidate, Algorithm 4.7 is used.

```
INPUT: Argument List
OUTPUT: semantic role candidate
For each argument i in argument list
    syntactic function i=get syntactic_function(i)// using Algorithm 4.3
    If syntactic function i = "Specifier"
            argument_list.remove(i)
    Else
            Semantic_role_candidate_list.add(i)
  End if
End for
```

Algorithm 4.7: Argument Identifier

The argument identifier takes argument list (returned by pruner) as an input then checking syntactic function then remove argument with syntactic function specifier and selects the other as possible semantic role candidate.

4.5.3 Feature Selection

Memory based learning learns new instances by storing previous training data into memory. The instances are extracted as features. When a new sentence constituent is given to be classified by the system, it accepts and de-constructs as instances to make similar representation with the one stored in memory. Therefore, the Feature selector accepts candidate arguments selected in argument identification stage along with their features as an input. It performs the following tasks to make argument similar representation with the one in memory. First it select the six feature for each candidate argument, second arrange the features in proper order similar to the one exist in learning model, third separate each features by comma and lastly it add question mark in the last index. It is suitable input for argument classification. For example the four instances illustrated in Figure 4.8 extracted from four argumnets of the sentence "ካሳ ሁለት ጊዜ በ ባንክ ለአስቴር ገንዘብ ልኮላታል /Kasa hulätəgize bä banəkə läʾäsəterə gänəzäbə ləkolatalə". The first for argument "ካሳ" and the second for argument "በ ባንክ" the third for argument "ለ አስቴር" and the last one for argument "ገንዘብ".

```
DABV, active, P, NP, #, SUBJ,   ?
DABV, active, bä, PP, O, MOD,    ?
DABV, active, lä, PP, P, IOBJ,    ?
DABV, active, M, PP, O, DOBJ,    ?
```

Figure 4.6: Instance for Unknown class of Arguments

Let's have a closer look at the values of first instance in Figure 4.8 (from left to right):

- **DABV:** Class of the verb "ልኮላታል/ləkolatalə" (send) it is ditransitive agentive benefactive verb class
- **Active:** Grammatical voice of the verb "ልኮላታል / ləkolatalə"

- **P:** Term category of the "ካሳ/kasa"
- **NP:** Phrase type of the constituent "ካሳ/kasa"
- **#:** Null because the argument node "ካሳ/kasa" is not PP
- **SUBJ:** Syntactic function of the argument "ካሳ/kasa"it is subject
- **? :** What is the target class?

4.5.4 Argument Classification

Classification in this stage assigns labels to the candidate arguments identified in the previous stage. When new sentence constituent needs to assign the semantic role, each possible constituent are similarly deconstructed and represented as instances and give to the classifier in order to classify.

An extrapolation is performed to assign the most likely neighborhood class. The extrapolation is based on the similarity metric applied on the training data. For this work Algorithm 4.8 used for argument classification task. The given instance is compared to each and every instance in the training set, recorded by the memory-based learner. In doing so, the classifier will try to find training instance in memory that most closely resembles it. If there is an exact match on the memory, the classifier returns (extrapolates) the class of that instance to the new instance. However, if there is no exact match of instances in the memory, it will use the selected metrics.

```
Input: Argument Instances, Concept Description
Output: semantic rolled arguments
Model=Load classification model
For each Instance  i in instance
    Concept Description i=crate concept_ description (i)
            For each class c in concept description
                    Similarity i, c =model.compute simlarity(concept_desc i,concept_desc c)
                    Simset.add(Similarity i, c)
            End for
        Class i=vote_class(simset)
End for
```

Algorithm 4.8 Argument classification

CHAPTER FIVE: EXPERIMENT AND RESULTS

In this chapter the experimental environment, algorithm parameter optimization, evaluation techniques, evaluation matrix and different experimentations are described. Besides, activities followed, and results obtained from the experiments are presented in detail.

5.1 The Corpus

In order to evaluate the performance of the model we conducted the experiment with 240 sentences which helps to show different verb class. From these sentences 551 instances (see Appendix D) were extracted and 21 different classes occur (see Appendix E). We are using all available data except one example as training material based on leave one out cross validation (LOO-CV) method.

5.2 Experimentation Environment

A memory based learning tool called Tilburg Memory-based Learner (TiMBL) is used for training of the classifier and testing of the classifier. TiMBL was designed with linguistic classification tasks in mind. It can in principle be applied to any kind of classification task with symbolic or numeric features and discrete (non-continuous) classes for which training data is available. TiMBL runs on Unix machine. The experiments were carried out using TiMBL version 6.0. Ubuntu 14.04 LTS used as working environment.

5.3 Algorithm Parameter Optimization

The “best” algorithm for some task using default settings may not necessarily be the best algorithm. A good choice of parameter settings can have a large effect on accuracy in TiMBL. Therefore, we tuned some of the parameters of the MBL algorithms to achieve the best performance on the given dataset. A good choice of parameter settings can have a large effect on accuracy. We tested our dataset on both default parameter and by tuning the different parameters. During our experiment the following algorithm parameters were optimized.

5.3.1 The Distance Metrics

There are different types of distance metrics including overlap and MVDM. As described in [36], the most basic metric that works for patterns with symbolic features is the Overlap metric which is the default metric in TiMBL. However, it is limited to exact matches between feature values. All values of a feature are seen as equally dissimilar. To optimize this we used modified value difference. Logically, we know that some feature values are more similar than others. Modified Value Difference Metric (MVDM) was defined to include this information about similar feature values. It determines the similarity of a feature by looking at the co-occurrence of values with target classes. The distance between two values $\mathbf{v_1}$ and $\mathbf{v_2}$ of a feature is computed through the difference of the conditional difference of the classes $\mathbf{C_i}$ for those values.

The way in which the nearest neighbor sets are composed in MVDM differs from the way in which it is done in Overlap-based metrics. It causes an abundance of ties in the nearest neighbor position; this problem is reducedd or completely resolved by using MVDM. For example, if a test instance differs with one mismatch from the nearest neighbor, Overlap will yield all the instances which have different feature values in the mismatch position as nearest neighbors. Contrary to this, MVDM will only propose the nearest neighbor with the lowest delta in the mismatch position. This means that MVDM yields a much smaller nearest neighbor set than the Overlap based metrics [36].

5.3.2 Feature Weighting Metrics

Feature weighting in supervised learning concerns the development of methods for quantifying the capability of features to discriminate instances from different classes [35]. The k-NN performance can be improved by identifying the significant features and finding the feature weights. It includes Gain ratio, Information gain, shared variance and chi-square and the last three parameters can be optimized [44]. A k-NN classifier in its most basic form operates under the implicit assumption that all features are of equal value as far as the classification problem at hand is concerned. When irrelevant and noisy features influence the neighborhood search to the same degree as highly relevant features, the accuracy of the model is likely to deteriorate. It is a technique used to approximate the optimal degree of influence of individual features using a training set. When successfully applied, relevant features are attributed a high weight value,

whereas irrelevant features are given a weight value close to zero. It can be used not only to improve classification accuracy but also to discard features with weights below a certain threshold value and thereby increase the resource efficiency of the classifier. Therefore, feature weighting is a measure, which assigns higher weights to more important features.

5.3.3 Number of Nearest-Neighbor

Optimizing of nearest neighbor representing the number of nearest distances in which memory items are searched. In the experiments, it varied between 1, 3, 5, 7, 9, 11, 15, 25, 35 and 45. Using a larger value of k can often lead to higher accuracy for discrete classes and larger training set. On the other hand, small value of k captures fine structure of problem space better and may be necessary with small training sets. Hence our training dataset is small. We examined our dataset by varying the value of k to 1 (default), 3 and 5 to see the effect of the k value in extrapolating the class of new instances and to get better generalization by setting all the parameters the same [35,44].

5.3.4 Optimization of the Class Voting Weight

The main task of the TiMBL tools beyond learning is determining the set of nearest neighbors. Once, the identification of nearest neighbor is performed, the output class can be decided using different ways such us normal majority voting, inverse distance weighting, inverse linear weighting, and exponential decay weighting.

The most straightforward method for letting the k-nearest neighbors vote on the class of a new case is the majority voting method. It is the default setting, in which the vote of each neighbor receives equal weight, and the class with the highest number of votes is chosen [35]. It is the basic method used to determine the class to be assigned to a query. It consists of doing a frequency count of the classes occurring in the nearest neighbor set, and then assigning the most frequently occurring class to the query. This means that all instances in the set of nearest neighbors contribute equally to the decision about which class to assign to the query, irrespective of their distances from the query point.

Table 5.1: Default Parameters with Their Values and Descriptions

Parameter	Default values	Description
Nearest neighbor	K=1	K=1 number of nearest neighbors used for extrapolation
Feature-weighting	-m0	The distance between two patterns is simply the sum of the differences between the features (overlap)
Distance metrics	-w0	No weight. All features have the same importance
Class voting	-dz	Normal majority voting-all neighbors have equal

5.4 Evaluation

The experiment started by executing TiMBL with the training files called *SRLA.train* together with selected algorithms and default metrics. Upon completion of the execution, a new file with the same training file name is created. The selected TiMBL algorithm and the metrics used during the experiment are appended to the new output file name. In addition to the result file, information about the operation of the algorithm is also sent to the standard output. Let us provide explanation what goes on the output. The output contains three phases. Figure 5.1 illustrate the three phases. Phase one is the training data analysis phase. Time stamps for start and end of analysis are provided. Some preliminary analysis of the training data is done: number of training items, number of classes, and entropy of the training data entropy in this case is a probability-based measure used to calculate the amount of uncertainty. As the data become purer and purer, the entropy value becomes smaller and smaller. The goal in machine learning is to get very low entropy in order to make the most accurate decisions and classifications. For each feature, the number of values, and four variants of an information-theoretic measure of feature relevance are given. These are used both for memory organization during training and for feature relevance weighting during testing. Finally, an ordering (permutation) of the features is given. This ordering is used for building the tree-index to the case-base.

Phase two is the learning phase: All training items are stored in an efficient way in memory for use during testing. Again timing information (real time) is provided, as well as information about

the size of the data structure representing the stored examples and the amount of compression achieved.

In phase three the trained classifier is applied to the test set. If we did not specify which algorithm to use the default algorithm are used (IB1 with information-theoretic feature weighting).

```
Examine datafile 'SRLA.train' gave the following results:
Number of Features: 6
InputFormat       : C4.5

Phase 1: Reading Datafile: SRLA.train
Start:         0 @ Thu Jun 1 05:46:45 2017
Finished:     551 @ Thu Jun 1 05:46:45 2017
Calculating Entropy        Thu Jun 105:46:45 2017
Lines of data     : 551
DB Entropy        : 3.6505580
Number of Classes : 21

Feats Vals              InfoGain            GainRatio
   1    14             1.0831090           0.29571687
   2     4             0.34768261          0.18016327
   3    46             2.2085100           0.48309625
   4     6             0.77923856          0.49613492
   5    24             1.7010695           0.68558160
   6     6             1.3132329           0.66234876
Preparation took 0 seconds, 4 milliseconds and 169 microseconds
Feature Permutation based on GainRatio/Values :
< 6, 4, 2, 5, 1, 3 >
Phase 2: Building multi index on Datafile: SRLA.train

Start:         0 @ Thu Jun 1 05:46:45 2017
Finished:     551 @ Thu Jun 1 14 05:46:45 2017
Phase 3: Learning from Datafile: SRLA.train
Start:         0 @ Thu Jun 1 05:46:45 2017
Finished:     551 @ Thu Jun 1 14 05:46:45 2017
Size of InstanceBase = 1402 Nodes, (56080 bytes), 57.92 % compression
Learning took 0 seconds, 7 milliseconds and 783 microseconds
Starting to test using Leave One Out
Writing output in:          SRLA.train.LOO.O.gr.k1.out
Algorithm     : LOO
Global metric : Overlap
Deviant Feature Metrics:(none)
Weighting     : GainRatio
Feature 1  : 0.295716871822004
Feature 2  : 0.180163265681561
Feature 3  : 0.483096245089483
Feature 4  : 0.496134916651934
Feature 5  : 0.685581595136088
Feature 6  : 0.662348756499878
```

Figure 5.1: Learning and Test Phases of the TiMBL output

5.4.1 Evaluation Technique

Simply splitting the corpus into a single training and testing set may not give the best estimate of future performance. Typically, classifiers are trained on a train data set and tested on a separate data set. Normally classifiers use 10% of the complete data set as test data and the remaining data as training data. This approach is fine when the training and test sets contain enough examples to get reliable results. Our data set of annotated sentences, however, is quite small training set with a small number of examples and can cause over fitting. Over fitting occurs when a model does not fit future states [45]. That is, the classifier is able to classify the train instances accurately but does not perform well on new data. To overcome this data scarcity problem, we trained the classifier using the leave one out cross-validation (LOOCV) techniques.(***-t leave_one_out*** option in TiMBL). With this option set, every data item in turn is selected once as a test item, and the classifier is trained on all remaining items. Accuracy of the classifier is then the number of data items correctly predicted. Using LOOCV guaranties that the classifier will be tested for every argument type and maximizes the amount of training instances available. This makes it an ideal method for training and testing classifiers with a limited amount of data.

Leave-one-out cross-validation (LOOCV)

Leave-one-out is one of the cross validation techniques, which uses all available data except one **(n-1)** example as a training material. It tests the classifier on the one held-out example by repeating it for all examples. In other words in each iteration nearly all the data except for a single observation are used for training and the model is tested on that single observation. An accuracy estimate obtained using LOOCV is known to be almost unbiased but it has high variance, leading to unreliable estimates. This method is useful when there are few samples in the dataset.

5.4.2 Evaluation Metrics

The correctness of a classification can be evaluated by computing the number of correctly recognized class examples (true positives), the number of correctly recognized examples that do not belong to the class (true negatives), and examples that either were incorrectly assigned to the class (false positives) or that were not recognized as class examples (false negatives) [46].

The evaluation metrics commonly used in Classification are Precision, Recall and F-score.

Precision: The number of correctly classified positive examples divided by the number of examples labeled by the system as positive.

Recall: The number of correctly classified positive examples divided by the number of positive examples in the data. F-score is a combination of the above.

Given a class ***C*** and an instance ***i,*** classification of ***i*** can be described in four ways [47]
True positive (TP): ***i*** is predicted to be in ***C*** by the classifier and is actually in it.
False positive (FP): ***i*** is predicted to be in ***C*** by the classifier but is actually not in it.
True negative (TN): ***i*** is not predicted to be in ***C*** by the classifier and is not actually in it.
False negative (FN): ***i*** is not predicted to be in ***C*** by the classifier but is actually in it.
To compute class-based performance measures we need two additional numbers: the total number of positive examples *P (P = TP +FN)* and negative examples *N (N = FP +TN).*

With these numbers we can compute: $Precision = \frac{TP}{TP+FP}$, the number of times the classifier has correctly made the decision that some instance has class **C**, proportional to the total number of times the classifier has assigned class **C**.

$Recall = \frac{TP}{FP+FN}$ also called true positive rate (TPR). The proportional number of times an instance i with class C in the test data has indeed been classified as class C by the classifier.

These measures are class-based; they tell something about the performance of the classifier per class. To measure the performance on the full set, sometimes averaging methods for F-scores are used. TiMBL can compute two different F-score averages: micro-average and macro-average. Micro-average is the weighted average, i.e., each class F-score is weighted proportionally to the frequency of the class in the test set. Macro-average is the un weighted average. An additional performance measure for classification systems other than the above is accuracy. Classification accuracy is determined by the percentage of instances placed in the correct class.

5.4.3 Test Result

Based on the default and optimized values of parameters of IB1 algorithm we obtained the result depicted on Table 5.2. Testing was done using LOO-CV technique with TiMBL. IB1 algorithms were selected for our task.

Table 5.2: Average precision, recall, and F-score of LOO-CV Using IB1- Algorithm

No of instance	Parameter	Accuracy	Precision	Recall	F-Score
550	Default	82.51%	80.83%	77.01%	78.19%
550	Optimized	89.29%	82.20%	78.54%	79.77%

From Table 5.2 we can see that optimizing some of the parameters results better achievement. The result is achieved with small dataset (550 instances). In addition to average value each class values are shown on Table 5.3 and Table 5.4 with default and optimized parameter respectively. We also draw the complete confusion metrics to see correctly classified and misclassified instances.

Table 5.3: Individual class performance with Default parameter

Class	Precision	Recall	F-score
ARG1-TEM	0.77931	0.84962	0.81259
ARG2-LOC	0.40000	0.57143	0.47059
ARG2-ATR	1.00000	1.00000	1.00000
ARGM-MNR	1.00000	0.93548	0.96667
ARG2-EXT	0.66667	0.66667	0.66667
ARG0-AGT	0.67857	0.63333	0.65517
ARG2-BEN	0.77778	0.58333	0.66667
ARG1-PAT	0.71429	0.62500	0.66667
ARGM-TMP	0.96907	0.97917	0.97409
ARGM-COM	0.91667	0.94286	0.92958
ARGM-LOC	0.90625	0.87879	0.89231
ARG2-INS	1.00000	0.85714	0.92308
ARG4-DES	0.79167	0.86364	0.82609
ARG3-SRC	0.66667	0.57143	0.61538
ARGM-CAU	1.00000	0.90000	0.94737
ARG2-DES	1.00000	0.66667	0.80000
ARGM-PUR	1.00000	0.75000	0.85714
ARG3-INS	0.66667	1.00000	0.80000
ARG0-EXP	0.66667	0.47059	0.55172
ARG1-EXP	0.37500	0.42857	0.40000
ARGM-NEG	1.00000	1.00000	1.00000

Table 5.4: Individual class performance with Optimized Parameter

Class	Precision	Recall	F-score
ARG1-TEM	0.91111	0.92481	0.91719
ARG2-LOC	0.50000	0.57143	0.53333
ARG2-ATR	1.00000	1.00000	1.00000
ARGM-MNR	1.00000	0.93548	0.96667
ARG2-EXT	0.80000	0.66667	0.72727
ARG0-AGT	0.78125	0.83333	0.80645
ARG2-BEN	0.66667	0.66667	0.66667
ARG1-PAT	0.85714	0.75000	0.80000
ARGM-TMP	0.96907	0.97917	0.97409
ARGM-COM	0.91667	0.94286	0.92958
ARGM-LOC	0.90909	0.90909	0.90909
ARG2-INS	1.00000	0.85714	0.92308
ARG4-DES	0.76000	0.86364	0.80851
ARG3-SRC	0.83333	0.71429	0.76923
ARGM-CAU	1.00000	0.90000	0.94737
ARG2-DES	0.33333	0.33333	0.33333
ARGM-PUR	1.00000	0.50000	0.66667
ARG3-INS	0.80000	1.00000	0.88889
ARG0-EXP	0.68750	0.64706	0.66667
ARG1-EXP	0.53846	0.50000	0.51852
ARGM-NEG	1.00000	1.00000	1.00000

Some observations can be made regarding the class-based results showed on Table 5.4:

- A sharp drop in recall for the some numbered arguments can be observed: recall for ARG2-LOC is 0.57143, recall for ARG2-DES is only 0.33333and recall for ARG1-EXP is only 0.50000. This can be contributed in part to the low number of training examples with these labels in the training dataset.
- The argument label with the highest F-score is ARG2-ATR and ARGM-NEG. This is probably due to the fact that the only information needed to assign this labels are the head word term category feature + phrase type of constituent, which makes classification of ARG2-ATR and ARGM-NEG relatively easy.

Table 5.5: Confusion metrics on IB1 optimized parameters with LOOCV

	ARG1-TEM	ARG2-LOC	ARG2-ATR	ARGM-MNR	ARG2-EXT	ARG0-AGT	ARG2-BEN	ARG1-PAT	ARGM-TMP	ARGM-COM	ARGM-LOC	ARG2-INS	ARG4-DES	ARG3-SRC	ARGM-CAU	ARG2-DES	ARGM-PUR	ARG3-INS	ARG0-EXP	ARG1-EXP	ARGM-NEG
ARG1-TEM	123	0	0	0	0	3	2	1	0	0	0	0	0	0	0	0	0	0	4	0	0
ARG2-LOC	0	4	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0
ARG2-ATR	0	0	4	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
ARGM-MNR	0	0	0	29	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
ARG2-EXT	1	0	0	0	4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
ARG0-AGT	2	0	0	0	0	50	1	1	0	0	0	0	0	0	0	0	0	0	1	5	0
ARG2-BEN	2	0	0	0	0	2	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ARG1-PAT	2	0	0	0	0	1	0	12	0	0	0	0	0	0	0	0	0	0	0	1	0
ARGM-TMP	0	0	0	0	0	0	0	0	94	2	0	0	0	0	0	0	0	0	0	0	0
ARGM-COM	0	0	0	0	0	0	0	0	2	33	0	0	0	0	0	0	0	0	0	0	0
ARGM-LOC	0	3	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0
ARG2-INS	0	0	0	0	0	0	0	0	1	0	0	6	0	0	0	0	0	0	0	0	0
ARG4-DES	0	1	0	0	0	0	0	0	0	0	0	0	19	0	0	1	0	0	0	0	0
ARG3-SRC	0	0	0	0	1	0	0	0	0	0	0	0	1	5	0	0	0	0	0	0	0
ARGM-CAU	0	0	0	0	0	0	0	0	0	0	1	0	0	0	18	0	0	1	0	0	0
ARG2-DES	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0	0
ARGM-PUR	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	2	0	0	0	0
ARG3-INS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
ARG0-EXP	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0
ARG1-EXP	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0
ARGM-NEG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20

From confusion matrix depicted in Table 5.5 we can observe that all off-diagonal elements on the confusion matrix represent misclassified data. Example ARG0-AGT is mis-classified three times as ARG1-TEM, two times as ARG2-BEN, one times as ARG1-PAT, one time as ARG0-EXP and six times as ARG1-EXP.

5.5 Discussion

The results reported here provide a first insight into the possibilities semantic role labeling for Amharic text based on memory based learning.

The experiments were carried out with the TiMBL software. Regarding the learning algorithm, the IB1 classifier is used, parameterized by default and using MVDM as the similarity metric, information gain for feature weighting, using 3k-nearest neighbors, and final decision as a function of their distance. We used four measures for the evaluation of our proposed system: accuracy, precision, recall and F-Score (a combined measure). The evaluation result showed that the accuracy in classifying semantic role achieved 82.51% accuracy and 78.19% F-Score with default parameter and 89.29% accuracy and 79.77% F-Score with optimized parameter setting for semantic role labeling task.

The learning system performed reasonably well on the selected feature set and manually annotated data. It showed a good performance by integrating IB1 with different parameter setting than IB1 default parameters.

Our proposed system is not compared with other SRL systems because the data format, data size and evaluation methods are all different. Furthermore, other systems were based on features extracted from dependency structures.

CHAPTER SIX: CONCLUSION AND FUTURE WORK

6.1 Conclusion

This study was conducted to develop SRL for Amharic text with a particular reference to simple sentences. We have developed three phases of semantic role labeling system for Amharic simple sentence using memory-based learning model. The first phase is text pre-processing phase, in this phase the input sentence words are tagged with their part of speech and parsed in order to make easier further processing. The second stage is training phase, in this phase parse tree is mapped to semantic role, then different features extracted from parse tree and pass to memory based learner to construct the classification model. The classification model helps to predict the semantic role of unknown constituent. The third phase is semantic role labeling, in this phase different arguments of sentence are identified along with different features from parse tree, then each argument features are arranged in the same format with the one stored in memory and pass to the classifier. The classifier predicts semantic role of the argument based on it learned from classification model.

The system yields promising result, 82.51% accuracy and 78.19% F-Score with default parameter and 89.29% accuracy and 79.77% F-Score with optimize parameter setting for semantic role labeling task.

Therefore, we can draw a number of conclusions from our investigation of SRL for Amharic text. First, memory based learning can successfully be applied to assign semantic role for Amharic. Second, basic syntactic and lexical features are useful for semantic role labeling. As well as attempting optimize TiMBL parameters significantly improve the system performance.

6.2 Contributions of the Study

The main contributions of the work are listed below.

- The general architecture of semantic role labeler for Amharic text using MBL is proposed.
- The system has developed and implemented feature extraction algorithms.
- The study has implemented classification algorithm used in other languages.

- The system has integrated morphological analyzer which was not used in previous works for semantic role labeling to extract feature.
- The system has integrated term category which was not used in previous works to prevent the storage of all head words in the memory.
- The system has implemented a memory based semantic role labeler for simple Amharic sentence.

6.3 Future Work

On the basis of the findings presented in Chapter 5, the following potential extensions that will enhance the performance, scope, and outcomes of the Amharic SRL are suggested as future works:

- We used PropBank, a semantic role labeling method and further study has to be tested the labeling with other methods such as FrameNet, merging approach, and abstract roles to get comparable results for Amharic SRL.
- In the context of this research, we didn't use verb sense as a feature. However, some verbs may have different senses which lead to different number of arguments. Thus, in future work, verb senses have to be used and tested as a feature which will improve the performance of the SRL.
- In machine learning, specifically classification tasks, reasonable size of corpus with balanced distribution of class are highly regulating the performance of the system. However, since we used small size of corpus which is not well balanced class distribution, in future it is required to test the performance of the proposed solution with larger and well balanced corpus.
- To test the performance of the system, LOO-CV is used as an evaluation technique and it works only with IB1 algorithm. Thus, in future other standard SRL evaluation techniques, including K-Fold cross validation have to be used to test the performance of the proposed solution.
- Language experts based semantically annotated corpus is used for the training of the role labeler and in future porting automatic parser will minimize manual efforts used in our proposed solution.

- In the scope of this research, simple sentence role labeling is targeted. However, due to existence of different kinds of sentences in Amharic it is required to extend the labeler for all kinds of sentence as a future work.
- Future research on area of frameset and PropBank for Amharic text need to be done.

REFERENCES

[1] F. Pascale, Z. Wu, Y. Yang and D. Wu, “Automatic Learning of Chinese English Semantic Structure Mapping”, *In Spoken Language Technology Workshop*, IEEE, 2006, pp. 203-233.

[2] A. Dan and J. Singh, “A Survey on Semantic Role Labeling and Dependency Parsing”, 2013.

[3] D. Jurafsky and J. H. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition", *Book Review*, University of Colorado, Boulder, 2015

[4] D. Shen, and M. Lapata, "Using Semantic Roles to Improve Question Answering" In *EMNLP-CoNLL*, pp. 12-21. 2007.

[5] G. Stevens, "Automatic Semantic Role Labeling in a Dutch Corpus", Unpublished Master Thesis, University Utrecht, 2007.

[6] I. Titov and A. Klementiev, "Semi-Supervised Semantic Role Labeling: Approaching from an Unsupervised Perspective", In *Proceedings of the COLING Conference*. 2012.

[7] D. Wu and P. Fung. Semantic Roles for SMT: A Hybrid Two-Pass Model. *In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 13–16.Association for Computational Linguistics, 2009.

[8] D. Liu and D. Gildea, “Semantic Role Features for Machine Translation", In *Proceedings of the 23rd International Conference on Computational Linguistics*, Association for Computational Linguistics, 2010, pp. 716-724.

[9] L. Marquez, X. Carreras, K. C. Litkowski and S. Stevenson, “Semantic Role Labeling: An Introduction to the Special Issue”, *Computational Linguistics*, 2008, pp. 145-159.

[10] D. Gildea and D. Jurafsky, "Automatic Labeling of Semantic Roles", *Computational Linguistics* Vol. 28, No. 3, 2002, pp. 245-288.

[11] S. Pradhan, W. Ward, K. Hacioglu, J. Martin and D. Jurafsky, ”Shallow Semantic Parsing using Support Vector Machines”, *In Proceedings of HLT/NAACL*, 2004, pp. 233-240.

[12] H. Sun and D. Jurafsky, “Shallow Semantic Parsing of Chinese”, In Proceedings of NAACL-HLT, Vol. 4, 2004, pp. 249-256.

[13] V. Hagen Fürstenau and B. Vorgelegt, "Semi-supervised Semantic Role Labeling via Graph Alignment", University at des Saarlandes, 2011.

[14] P. Exner and Pierre Nugues, "Using Semantic Role Labeling to Extract Events from Wikipedia", In *Proceedings of the Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011), Workshop in Conjunction with the 10th International Semantic Web Conference,* 2011, pp. 23-24.

[15] R. P. T. D. Santos, "Automatic Semantic Role Labeling for European Portuguese", *PhD Diss.,* 2014.

[16] T. Samardzic, "Semantic Roles in Natural Language Processing and in Linguistic Theory", Unpublished PhD Dissertation, Tesis de lic., Universidad de Ginebra, 2009.

[17] M. Aronoff and K. Fudeman, "What is Morphology?", *Blackwell Publishing*, Vol. 8, 2011.

[18] Baye Yimam, "የአማርኛ ሰዋስው", *EMPDA*, Addis Ababa, 1987 (EC).

[19] A. Al-Taani, M. Msallam, and S. Wedian, "A Top-Down Chart Parser for Analyzing Arabic Sentences", *The International Arab Journal of Information Technology,* Vol. 9, No. 2, 2012, pp. 109-115.

[20] M. A. Covington, "A fundamental algorithm for dependency parsing." *Proceedings of the 39th annual ACM southeast conference*. 2001.

[21] Fillmore, Charles J., Russell Lee-Goldman, and Russell Rhodes. "The framenet constructicon." *Sign-based construction grammar* (2012): 309-372.

[22] M. Palmer, D. Gildea and P. Kingsbury, "The Proposition Bank: An Annotated Corpus of Semantic Roles", *Computational Linguistics,* Vol. 31, No. 1, 2005, pp. 71–105.

[23] D. Dowty, "Thematic Proto-roles and Argument Selection", *Language,* Vol. 67, No. 3, pp. 547–619.

[24] Xue, N. and Palmer, M.,"Calibrating features for semantic role labeling", *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP),*2004, pp. 88-94

[25] C. Bonial, O. Babko-Malaya, J. D. Choi, J. Hwang and M. Palmer, "Propbank Annotation Guidelines", *Center for Computational Language and Education Research Institute of Cognitive Science University of Colorad at Boulder,* 2010.

[26] C. Bonial, J. Hwang, J. Bonn, K. Conger, O. Babko-Malaya, and M. Palmer, “English Propbank Annotation Guidelines”, *Center for Computational Language and Education Research Institute of Cognitive Science University of Colorado at Boulder*, 2012.

[27] S. Wermter, EllenRilo and Gabriele Scheler, “Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing”, Vol. 1040, *Springer Science & Business Media,* 1996.

[28] K. Shaalan, “Rule-based Approach in Arabic Natural Language Processing”, *The International Journal on Information and Communication Technologies (IJICT),* Vol. 3, No. 3, 2010, pp. 11-19.

[29] C. D. Manning and H. Schütze, “Foundations of Statistical Natural Language Processing”, Vol. 999. Cambridge: MIT press, 1999.

[30] R. Swier and S. Stevenson, "Unsupervised Semantic Role Labeling" *In Proceedings of EMNLP*, 2004.

[31] W. Daelemans, and A. Van den Bosch, “Memory-based Language Processing”, *Cambridge University Press*, 2005.

[32] Daelemans, Walter, Antal Van Den Bosch, and Ton Weijters. "IGTree: using trees for compression and classification in lazy learning algorithms." *Artificial intelligence review* 11, no. 1-5 (1997): 407-423.

[33] W. Daelemans, A. van den Bosch, and J. Zavrel, “A Feature-relevance Heuristic for Indexing and Compressing Large Case Bases” In *Poster Papers of the Ninth European Conference on Machine Learning*, 1997, pp. 29-38.

[34] D. Aha, J. Kibler and J. Albcrt, “Instance based Algorithms in Machine learning”, Vol. 6, 1991, pp. 37-66.

[35] W. Daelemans, J. Zavrel, K. van der Sloot and A. Van den Bosch ,“Timbl: Tilburg Memory-Based Learner", *Tilburg University* , 2004.

[36] W. Daelemans, J. Zavrel, K. van der Sloot and A. van den Bosch, “TiMBL: Tilburg Memory-Based Learner version 5.0 Reference Guide: ILK Technical Report”, ILK 03-10 http://ilk.uvt.nl/timbl/,2003.

[37] ጌታሁን አማረ “ዘመናዊ የአማርኛ ሰዋስው በቀላል አቀራረብ”, Addis Ababa, 1989 (EC).

[38] V. Punyakanok, D. Roth, and W. Yih, “The Importance of Syntactic Parsing and Inference in Semantic Role Labeling” *Computational Linguistics* Vol. 34, No. 2, 2008, pp. 257-287.

[39] X. Wang, “Semantic Role Labeling in Chinese Using HowNet”, *Language and Linguistics*, Vol. 9, No. 2, 2008, pp. 449-461.

[40] A. K. Ghalibaf, S. Rahati and A. Estaji, “Shallow Semantic Parsing of Persian Sentences”, *PACLIC*, 2009.

[41] K. Wang, “Automatic Semantic Role Labeling for Chinese”, *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Vol. 3, 2010, pp. 182-185.

[42] J. Teixeira, C. Oliveira and C. Moutinho “Machine Learning of European Portuguese Grapheme-To-Phone Conversion using a Richer Feature Set”, Vol. 4, No. 6, 2006.

[43] D.W. Aha, Dennis Kibler, and Marc K. Albert. "Instance-based learning algorithms." *Machine learning* 6.1 (1991): 37-66.

[44] E. Marchiori, Class Dependent Feature Weighting and K-nearest Neighbor Classification, In *IAPR International Conference on Pattern Recognition in Bioinformatics*, Springer Berlin Heidelberg, 2013, pp. 69-78.

[45] M. H. Dunham, “Data Mining: Introductory and Advanced Topics*”, Pearson Education India*, 2006.

[46] M. Sokolova and G. Lapalme, “A Systematic Analysis of Performance Measures for Classification Tasks”, *Information Processing and Management*, Vol. 45, 2009, pp. 427–437.

[47] Margaret H. Dunham, “Data mining, introductory and advanced topics”, Pearson Education Inc., Upper Saddle River, New Jersey, USA 2003.

[48] M. Gasser, "HornMorpho: A System for Morphological Processing of Amharic, Oromo, and Tigrinya", In *Conference on Human Language Technology for Development, Alexandria, Egypt*, 2011.

APPENDIX

Appendix A: List of part of speech tags and their Description

No	POS name	Description
1.	N	Noun
2.	ADJ	Adjective
3.	ADV	Adverb
4.	P	Preposition
5.	Det	Determiner
6.	NUMCR	Cardinal number
7.	NUMOR	Ordinal number
8.	EV	Existence verb
9.	AV	Attributive verb
10.	SV	Scalar verb
11.	BV	Benefactive verb
12.	EXV	Expirencer verb
13.	TCV	Transitive-causative verb
14.	TAV	Transitive-agentive verb
15.	DALV	Ditransitive-agentive-locative verb
16.	DABV	Ditransitive-agentive-benefactive verb
17.	UMV	Unaccusatives-motion verb
18.	USV	Unaccusatives-state verb
19.	AUV	Agentive-Unergative verb
20.	EUV	Expirencer-Unergative verb
21.	PV	Possessive verb
22.	NP	Noun phrase
23.	VP	Verb phrase
24.	ADJP	Adjective phrase
25.	ADVP	Adverbial phrase
26.	PP	Prepositional phrase
27.	UNC	Unclassified
28.	PRON	Pronoun
29.	CONJ	Conjuction
30.	PRONP	pronoun phrase

Appendix B: Verb Class and their Semantic Classes

No	Verb class	Class name	Semantic Role Properties	Example verb
1.	Existence	EV	[+Agent, location]	አለ
2.	Attributive	AV	[+Agent, +attribute]	ሆነ
3.	Scalar	SV	[+Agent, +extension]	አደገ
4.	Benefactive	BV	[+experiencer, + theme]	ይወዳታል፣ ይጠላታል
5.	Experiencer	EXV	[+Agent, +experiencer]	አሳዘነ ፣ አስደሰተ
6.	Transitive-causative	TCV	[+Agent,+patient, inst]	ሰበረ ፣ቆረጠ
7.	Transitive-agentive	TAV	[+Agent,+theme]	ጎበኘ፣አገኘ፣ገዛ
8.	Possessive	PV	[+theme, +Agent]	አለው፣አላት
9.	Ditransitive-agentive-locative	DALV	[+Agent,+ theme, location]	አስቀመጠ
10.	Ditransitive-agentive-benefactive	DABV	[+Agent,+ theme, beneficiary,]	ሰጠ፣ለገሱ
11.	Unaccusatives-motion	UMV	[+theme, (location, destination, source)]	ሄደ፣መጣ
12.	Unaccusatives-state	USV	[+theme]	በቀለ፣ነፃ
13.	Agentive- Unergative	AUV	[+Agent]	ተራመደ፣ሮጠ
14.	Expirencer- Unergative	EUV	[+experiencer]	ሳቀ፣ተደሰተ፣ ተከፋ

Appendix C: Sample Parse tree with semantic role layer in string format

('S(ARG0-AGT (NP(N ካሳ)))(VP(ARG1-PAT(NP(N ምሳውን)))(Rel(TCV በላ)))')

('S (ARG0-AGT(NP (NDet የትምህርት)(N ሚኒስቴር)))(VP (ARG2-BEN (PP (P ለ)(NP(N ትምህርት))))(ARG1 TEM(NP (NUM 10.8 ሚሊየን) (N ብር))) (REL(DABV መደበ)))')

('S (ARG0-AGT(NP (NDet የኢትዮጲያ) (N ንግድባንክ)))(VP (ARGM-LOC(PP (P በ)(NP (NP(N አዲግራት))(N ከተማ)))) (ARGM-TMP (N ጥር 20)) (VP (ARG1-TEM(NP (AdjP (Adj አዲስ))(NDet የቅርንጫፍ)(N ቢሮ)))(REL(TAV ከፈተ))))')

('S(ARG0-AGT(NP(N ኮከቦች)))(VP(ARGM-TMP(PP(P በ)(NP(N ንጋት))))(ARGM-MNR(ADVP(ADV በጣም))) (REL(USV ያበራሉ))) ')

('S (ARG0-AGT(NP(N አስቴር)))(VP(ARG2-BEN(PP(P ለ)(NP(N ካሳ))))(ARG1-TMP(NP(N መፅሀፍ)))(REL(DABV ሰጠችዉ)))')

('S (ARG1-TEM(NP(N አሳው)))(VP(ARG3-SRC(PP(P ከ)(NP(N ባህር))))(REL(UMV ወጣ)))')

('S (ARG0-AGT(NP(N ካሳ)))(VP(ARG2-INS(PP(P በ)(NP(N ጦር))))(VP(ARG1-TEM(NP(N አንበሳ)))(REL(VT ገደለ))))')

('S (ARG0-AGT(NP(N ካሳ)))(VP(ARGM-MNR(ADVP(PP(P እንደ)(NP(N አባቱ)))(ADV ክፍኛ)))(VP(REL(V ታመመ)))) ')

('S (ARG0-AGT(NP(N ካሳ)))(VP(ARGM-COM(PP(P ከ)(NP(N ወንድሙ))))(VP(ARG4-DES(PP(P ወደ)(NP(N ጎጃም))))(REL(VM ሄደ)))) ')

('S (ARG0-AGT(NP(NP (N ሁለት))(NP(ADJP(ADJ ትልልቅ))(NP(N ልጆች)))))(VP(ARGM-TMP(NP(N ትናንት)))(ARG2-INS(PP(P በ)(NP(N መኪና))))(VP(ARG4-DES(PP(P ወደ)(NP(N ጎጃም))))(REL(VM(ሄዱ)))))')

('S(ARG0-AGT(NP(N አደጋው)))(VP(ARG1-EXP(NP(det የ)(NP(NP(N ኢትዮጵያን))(N ህዝብ))))(REL(V(አሳዘነ)))')

Appendix D: Sample Instance that Retrieved from parse tree with semantic role layer

TCV,Active,P,NP,#,SUBJ,ARG0-AGT
TCV,Active,MA,NP,#,OBJ,ARG1-PAT
TCV,Active,CL,NP,#,OBJ,ARG1-PAT
DABV,Active,EVT,NP,#,DOBJ,ARG1-TEM
DABV,Active,CL,NP,#,DOBJ,ARG1-TEM
DABV,Active,TA,NP,#,DOBJ,ARG1-TEM
DABV,Active,FR,NP,#,DOBJ,ARG1-TEM
DABV,Active,FO,NP,#,DOBJ,ARG1-TEM
DABV,Active,lä,PP,ANI,IOBJ,ARG2-BEN
DABV,Active,lä,PP,FR,IOBJ,ARG2-BEN
DABV,Passive,bä,PP,O,IOBJ,ARG0-AGT
DABV,Passive,ANI,NP,#,SUBJ,ARG2-BEN
DABV,Passive,EVT,NP,#,DOBJ,ARG1-TEM
DABV,Passive,CL,NP,#,DOBJ,ARG1-TEM
TAV,Active,EVT,NP,#,OBJ,ARG1-TEM
TAV,Active,T,ADVP,#,MOD,ARGM-TMP
TCV,Active,Bä,PP,T,MOD,ARGM-TMP
TCV,Active,L,NP,#,MOD,ARGM-LOC
TCV,Active,Kä,PP,garə,MOD,ARGM-COM
TCV,Active,garə,PP,P,MOD,ARGM-COM
TCV,Active,Bä,PP,CAU,MOD,ARGM-CAU
TCV,Active,MNR,ADVP,#,MOD,ARGM-MNR
TCV,Active,Bä,NP,MNR,MOD,ARGM-MNR
DABV,Active,P,NP,#,SUBJ,ARG0-AGT
DABV,Active,P,PRONP,#,SUBJ,ARG0-AGT
DABV,Active,lä,PP,P,IOBJ,ARG2-BEN
DABV,Active,INS,NP,#,DOBJ,ARG1-TEM
DABV,Active,O,NP,#,SUBJ,ARG0-AGT
DABV,Active,lä,PP,O,IOBJ,ARG2-BEN
DABV,Active,M,NP,#,DOBJ,ARG1-TEM
TCV,Active,P,PRONP,#,SUBJ,ARG0-AGT
TCV,Active,P,NP,#,OBJ,ARG1-PAT
TCV,Active,P,PRONP,#,OBJ,ARG1-PAT
TCV,Active,O,NP,#,SUBJ,ARG0-AGT
TCV,Active,O,NP,#,OBJ,ARG1-PAT
TCV,Active,ANI,NP,#,SUBJ,ARG0-AGT
TCV,Active,ANI,NP,#,OBJ,ARG1-PAT

Appendix E: List of Propbank Semantic Role Used in the System

No	Semantic Role	Description
1.	ARG0-ARG	Agent
2.	ARG1-TEM	Theme
3.	ARG1-PAT	Patient
4.	ARG0-EXP	Experiencer
5.	ARG1-EXP	Experiencer
6.	ARG1-EXT	Extension
7.	ARG2-BEN	Beneficiary
8.	ARG2-INS	Instrument
9.	ARG3-INS	Instrument
10.	ARG2-ATR	Attribute
11.	ARG3-SRC	Source
12.	ARG4-DES	Direction or Goal
13.	ARGM-DES	Destination
14.	ARGM-PUR	Purpose
15.	ARGM-CAU	Cause
16.	ARGM-LOC	Location
17.	ARGM-TMP	Time
18.	ARGM-MNR	Manner
19.	ARGM-PUR	Purpose
20.	ARGM-COM	Comitative
21.	ARGM-NEG	Negation

DECLARATION

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

Declared by:

Name: _____

Signature: _____

Date: _____

Confirmed by advisor:

Name: _____

Signature: _____

Date: _____