ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

Application of Part-of-Speech Tagged Corpus to improve the Performance of Word Sense Disambiguation: the case of Amharic

BIRUK RETTA

JUNE, 2015

Declaration

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

_____

Biruk Retta

This thesis has been submitted for examination with my approval as university advisor.

Dr. Solomon Tefera

Advisor

June, 2015

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION SCIENCE

Application of Part-of-Speech Tagged Corpus to improve the Performance of Word Sense Disambiguation: the case of Amharic

A Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Science

By:
BIRUK RETTA

JUNE, 2015

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
SCHOOL OF INFORMATION SCIENCE

Application of Part-of-Speech Tagged Corpus to improve the Performance of Word Sense Disambiguation: the case of Amharic

By:
BIRUK RETTA

Name and signature of Members of the Examining Board

Name	Title	Signature	Date
_____	Chair Person,	_____	_____
_____	Advisor,	_____	_____
_____	Examiner (s),	_____	_____
_____	Examiner (s),	_____	_____

ACKNOWLEDGMENT

First of all, I would like to thank God to be with me not only during this thesis but throughout my life. Next I owe a considerable debt and sincere thanks to my advisor Dr. Solomon Tefera for initiating this research idea, for providing me his POS tagger to be used during this research, for his valuable ideas, critical comments and supportive advices since the inception to the completion of this thesis. I have learnt humanity beyond the academic advises from him and I would like to say thank you from the bottom of my heart. In addition, I really thank Getahun Wassie for letting me to use the corpus he has prepared during his experiment.

My foremost gratitude goes to my father **Ato Retta Worku** and my mother **W/o Tsehaye Getaneh**, both are my inspirations since my childhood. Words may seem feeble to express my gratitude to you Mom and Dad; I would like to say thanks from deep inside my heart for your unreserved effort to help me at each part of my life since my birth. It is my pleasure to say you long-live, Mom and Dad!

I am also indebted to my dearest friends Dejene Hundessa , Birhanu Herano, Ashenafi Girma and Aschalew Belay for sharing ideas and different resources since the start to the end of this graduate study.

In addition, I can find no word to express my heartfelt special thanks to Dr. Million Meshesha, who has given me the course Analysis of algorithm during my undergraduate study and the courses Workshop and Data Mining during my graduate study for his dedication, tireless and unreserved effort to let me understand and visualize the course he has given.

Finally I would like to give my gratitude to people whose names are not mentioned but whose effort helped me much all along.

DEDICATED

TO

My Family

Table of Contents

List of Figures

List of Tables

LIST OF APPENDICES

LIST OF ACRONYMS

WSD	Word Sense Disambiguation
NLP	Natural Language Processing
MT	Machine Translation
MRD	Machine Readable Dictionaries
IE	Information Extraction
QA	Question Answering
IR	Information Retrieval
FDRE	Federal Democratic Republic of Ethiopia
HLT	Human Language Technologies
AI	Artificial Intelligence
OALD	Oxford Advanced Learner's Dictionary
EM	Expected Maximization
SA	Sentiment Analysis
SP	Shallow Parsing
NER	Named Entity Recognition
TE	Text Entailment
CLIR	Common Language Information Retrieval
SVM	Support Vector Machine
KNN	K-nearest Neighbor
SOV	Subject Object Verb
BNC	British National Corpus
POS	Part of Speech
TnT	Trigrams' n Tags

MBT Memory Based Tagger

HMM Hidden Markov Model

SMO Sequential Minimal Optimization

ABSTRACT

Natural language inherently involves polysemy, words which can be interpreted in multiple ways depending on the context in which they occur. Even if, Human brain is capable of identifying sense of a polysemous word spontaneously from a given context; ambiguity in natural language is a hindrance for users to utilize information technology to the fullest. Hence, it is of paramount importance to handle it computationally. Word Sense Disambiguation, one of the open research area in NLP, is a task focused on figuring out the intended meaning of a polysemous word in context. Thus, this study has focused on investigation of the application of POS tagged corpus on the performance improvement of WSD.

During the study, a corpus based approach was used involving supervised, unsupervised and semi-supervised machine learning paradigms. Five ambiguous Amharic words: bela, tenesa, derese, ale and *eTena* with about 1031 sentences involving two senses of each ambiguous word were used after adding POS tag to each word involved in the text corpus. Besides, two unsupervised algorithms (EM and Simple K-means) and five classification algorithms (AdaboostM1, Bagging, ADtree, SMO and Naïve Bayes) were used. Among the three machine learning paradigms, semi-supervised has achieved a score of 92.66% using ADtree, 92.33% using AdaboostM1, 89.92% using SMO, 80.98% using Bagging and 60.62% using Naïve Bayes algorithm. In addition, one seed word has been found to result better accuracy for WSD research using the above mentioned algorithms. The optimal average window size of 6-6 has been considered enough while POS tag information is involved for WSD study in Amharic.

So, for WSD study in Amharic using semi-supervised machine learning paradigm; inclusion of POS tag information to each word in the corpus has been found to yield better performance improvement of 4.2% using ADtree, 8.4% using AdaboostM1, 1.1% using Bagging, 2.5% using SMO and 12.6% using Naïve Bayes algorithm than the performance score of the baseline. Lastly, the researcher recommends further researches to be conducted for other ambiguous words and using different approaches to better address a problem of WSD.

Keywords: Ambiguity, Machine Learning Paradigms, NLP, POS Tagged Corpus, Polysemy, WSD

CHAPTER ONE

1. INTRODUCTION

1.1 Background of Word Sense Disambiguation

These days, the growth of information technology has paved the way for a sheer volume of information to be available for the society. Discussion about importance of a language for using the available information is not far from obvious since it serves as a medium of communication among the races. Language has a potential to express a wide range of ideas and to convey complex thoughts. In particular, natural language is now being used to exchange information among humans and has now reached to the extent of being evolution criteria for technology [1].

In order for the available information to be useable by the society, a need has emerged to make use of technology to process Natural Language. In response to such a need, NLP has come up with an indispensible focus of natural language computations. NLP on its effort to facilitate human-machine interaction tries to enable computers to be used as an aid in analyzing and processing natural languages to the extent of giving computer systems the ability to generate and interpret natural languages [2]. It has recently reached a stage of maturity and is no longer confined to a research task focused on only simple examples. Nowadays, it has been used with real people talking on the phone to machines, newspaper articles being scanned and manuals being corrected by machines etc. [3].

Ambiguity, one of the greatest challenges in NLP, is the term used to refer for understanding of something in two or more possible ways or something which has more than one meaning. It can appear in sentence or clause level (called structural or syntactic ambiguity) or at a word level (called lexical ambiguity) and has been intertwined with human language since the rise of philological communication [4].

Ambiguity is a universally recognized linguistic phenomenon which arises from the structure of the language and can be explained in terms of the analysis at different levels [5]. When humans process natural language they commonly have capability to overlook the ambiguities which is hardly possible for machines.

In Natural Language, most words have more than one lexical meaning though only one of them is active in a given context (discourse) i.e. ambiguity is a key feature of natural languages. Human language is fairly ambiguous with the average number of ambiguities that exist in most languages being 2.3 senses per word [6].Those words which involve multiple meanings or senses are called polysemous words. Polysemous words can be described in several different ways based on the context in which they occur. The context of polysemous words will be certified by the other neighboring words which is called local or sentential context. For example if we consider the Amharic word 'atena', it involves three different senses and meanings depending on the context in which it is used. In one sense it is used to mean "STRENGTHEN", in some other sense it is used to mean "STUDYING" and it is also used to mean "STRONG STICK". As it is seen for the different meanings of the word 'atena', the correct sense of the word is made clear based on the context in which the word has been used. The process of disambiguation for the different meanings of polysemous word relies on the context of the target word and also analysis of the properties exhibited by that context. Hence, WSD as one of the tasks in NLP is concerned with computationally determining which sense of the ambiguous word has been used in a given context [2].

Lexical ambiguity involves syntactic or semantic ambiguity. Syntactic ambiguity can be solved with the use of POS taggers while semantic ambiguity requires WSD system to be disambiguated [7]. Lexical Disambiguation, which is generally called as WSD in the field of computational linguistics, is defined as a process of assigning the appropriate sense of a word in a specific context, from a set of predefined possibilities and is considered as a core for understanding. WSD, a process that appears largely to be unconscious for people, is also thought to be the key ingredient for the evolution and development of a language. Human beings (naturally intelligent) are capable to proficiently manage such disambiguities by

integrating a wide variety of semantic and syntactic hints cognitively and linguistically in addition to the incorporation of world knowledge and the surrounding context [8]. Computationally handling ambiguities by developing algorithms which can replicate the disambiguation capability of human beings become challenging task [9].

WSD is a long-standing problem in NLP and has been formulated as a distinct computational task during the early days of MT during the late 1940s. The problem has been introduced by Weaver in 1949 with his work on MT. Automatic sense disambiguation approaches lack world knowledge but has been a focus of attention since the earliest times where computers were used for treatment of a language in 1950s. Since then, different efforts have been carried out to develop WSD systems with the incorporation of MRD, thesaurus, and other knowledge-based resources in 1980s and corpus based techniques (machine learning approaches) during 1990s and 2000s.

WSD relies on knowledge as its vital component since the knowledge sources provide the data necessary to associate senses with words. The knowledge sources can generally be classified as structured like thesauri, MRD, ontologies etc. and unstructured as in the case of corpora (raw corpora and sense annotated corpora) and collocation resources[10][11].

The overall goal of WSD system is to attribute the correct sense of polysemous words in a given text (discourse) for successful disambiguation tasks. Basically, there are two approaches to tackle the problem of WSD in NLP namely shallow and deep approaches [12]. According to Preeti [12], Shallow approach focuses on considering the surrounding words rather than understanding the text. In this approach, rules of word sense are extracted automatically from a training corpus of words tagged with their word senses [13]. In the case of deep approach [14], access to a comprehensive world knowledge will be performed to determine in which sense a word is used though this approach is not useable practically due to lack of MRD particularly for under-resourced languages.

The three approaches used in WSD systems for the acquisition of disambiguation information are: Knowledge based, Corpus based and Hybrid. Knowledge based approaches exploit lexical resources stored in machine readable formats or thesaurus and avoid the need for sense annotated data, corpus based approaches on the other hand require semantically annotated

corpora to train machine learning algorithms (supervised, semi-supervised and unsupervised) and lastly hybrid approaches involve both knowledge based and corpus based approaches together to increase the performance of WSD systems since knowledge based systems can cope with lack of sense annotated data (if used together with supervised machine learning paradigm) [15].

WSD is considered as an intermediary step for many natural language applications rather than being used as an end by itself. Applications which incorporate WSD systems as one component are MT, IE, text processing, QA systems, IR, text mining etc. In addition, WSD systems are also considered to provide greater importance in recently emerging areas such as bioinformatics and semantic web [16].

WSD as a process for finding the appropriate sense of a target word is performed on a set of words. It can be thought as a classification task where each word can be classified in to its candidate senses. The two major variations of WSD are:

- ✓ Lexical Sampling (Target Word WSD): focuses on disambiguation of restricted set of target words. For the disambiguation task supervised systems are used after being trained for each of the manually tagged target word.
- ✓ All words WSD: focuses on the disambiguation of all content words in the given corpus. Due to the data sparseness problem faced by supervised systems, a wide coverage of knowledge based or unsupervised approaches are used for this task.

Of the above two, all word WSD is deemed more realistic form of evaluation unlike target word WSD. Target word WSD depend on tagging judgment which is to be done once for a block of instances of the target word. Even though all word WSD will bring greater accuracy, it is not feasible due to the high cost required for producing the corpus as it requires tagging judgment of annotators.

The focus of attention during the task of developing WSD prototype model is context of target word, an unstructured piece of text which preserves necessary information which can be

represented as the combination of various features like lema, POS-tag, morphology, semantic relations etc [17].

Due to the importance of WSD for understanding semantics and many real-world applications researchers have been diligently trying to tackle the problem. The existing research works focus on few words due to the reliance on manually annotated texts or limited performance results. Thus, it urges to go for broad coverage and enhanced disambiguation accuracy to get benefit from it while incorporated for real world NLP applications [18]. This study has also focused on investigating effect of using POS tag information for WSD tasks.

1.2 Statement of the problem

Nowadays, the tools and methods used for processing natural language are getting advanced through time. Particularly, well-resourced languages like English, Spanish, French, Germany from Europe and Chinese, Japanese from Asia were the focus of the advancement. Despite the well-resourced languages, languages in the developing countries like Amharic and others have larger demand for investigating the application of computational linguistic methods and tools (like MT, WSD, IR systems etc) and it urges to work a lot on the area [19].

As it has been described in the introduction part, WSD is an open research area in NLP since the start of the field due to the phenomenon of polysemy. The problem has been given wider attention due to the larger number of polysemous words; if we consider English language as an example about 40% of the semantically significant words are polysemous [15]. Hence, it is worth a lot to find a mechanism for the artificially intelligent machines invented by human beings to manage WSD issues [20].

As it has been addressed by [21], developing an accurate WSD system in general has been more difficult to achieve. Moreover, resolution of the ambiguity of words has also been challenging computationally due to its reliance on knowledge, its representation, extraction and analysis. In spite of the challenges, the accuracy has improved over time with the use of different approaches (knowledge based and corpus based approaches) for different languages like English, French, German etc. [9]. The current state of the art accuracy has reached to a range of

(60-70) % that depicts the need for several information sources and techniques for having full-fledged lexical ambiguity [22].

Of the efforts taking place for improving WSD systems, some researches have shown the importance of POS tag to yield better performance than using raw text corpora. As POS tagging is a classification task focused on assigning part of speech information like noun, pronoun, verb, adverb etc to each word in a given sentence automatically; it could serve as an input to a wide variety of language applications like WSD, MT etc [23]. According to [24], POS tag is of paramount importance to be involved in the corpus used for developing improved WSD system. In addition, [25] has also shown the incorporation of corpus with part-of-speech tag information to yield better performance on the WSD model built. The study has also considered the performance impact which could also be observed with the choice of a given part-of-speech tagger for a given WSD method and performance impact with the incorporation of lemmatizer and stemmer in addition to the tagging.

Amharic is a Semitic language predominantly spoken in the FDRE. Ethiopia is a country having a population size of over 90 million at present and is linguistically diverse [26].Despite the linguistic diversity in the country, Amharic is being used as a working language of the federal government of Ethiopia. The language is serving as a medium of instruction in primary schools, court system, on all official documents, in electronically published documents in the form of articles, newspapers, reports etc. In addition, the language has been used for information storage and media communication purposes in the country.

According to [27], many languages specifically those of the developing countries (like Amharic) lack sufficient resources and tools required for the implementation of HLT and are commonly called under-resourced or pi languages. To bring up under-resourced languages, future researches on multilingual applications deemed indispensible. Amharic as one of the under-resourced and morphologically complex languages lack sufficient resources for the development of natural language technologies. Due to the above resource limitations, WSD researches in Amharic have been carried out using small set of data collected by researchers.

Despite well-resourced languages, research works for Amharic is at the start due to unavailability of large-scale linguistic resources.

To the best of the researcher's knowledge the research works done on Amharic WSD are:-

Wube [28] has applied rule based parser for Amharic sentence Disambiguation. During the study, a performance score of 86% has been achieved while parsing to resolve the structural ambiguity despite small size Amharic sentences used for rule induction.

Teshome [29] has used the principles of semantic vector analysis to Amharic WSD to improve the performance of IR system and achieved 82% accuracy of WSD.

Solomon [30] has used supervised machine learning approach to tackle the problem of WSD with the use of Naïve Baye's algorithm though lack of sense tagged corpora was the main set back faced by the researcher.

Solomon [31] has applied unsupervised machine learning approach to resolve problem using untagged corpora and end up with lower WSD accuracy as compared to supervised approach.

Getahun [32] who has been motivated with the attractive results of current English WSD systems using semi-supervised machine learning algorithms, has used semi-supervised machine learning algorithms to tackle the problem of Amharic WSD and got encouraging result.

Hagerie [33] has applied Adaboost and Bagging ensemble classifiers to tackle the problem of WSD problem. The experiment has shown that ensemble classifier algorithms could have comparable performance score with that of Naïve Bayes and K-nearest neighbor algorithms.

The other effort to build WSD prototype for Amharic was by Birhanie [34], ensemble of Naïve Bayes has been used to tackle the problem though end up with less accuracy unlike the semi-supervised approach.

Of the above mentioned research works in Amharic texts, a semi-supervised machine learning approach used by [32] has resulted better accuracy due to the bootstrapping nature of machine learning approach. In addition, Getahun [32] has recommended further research work to be done using the corpus prepared by him for taking the advantage of POS tag information to add value for the efforts to build full-fledged WSD system.

So, the above justification of the problem could have posed keen interest to this study as it could be used as an input to boost the performance of wider applications of language

technologies. Nevertheless, as to the reach of the researcher's knowledge, the researcher does not come across any literature that bears the idea of application of Part-of speech tag corpus for improving the performance of WSD for Amharic texts. Hence, the problem has been paid due attention by this study to address the problem of ambiguity at a word level for Amharic texts, thus series of experiments have been conducted in this study to come up with the application of Part-of-speech tagged corpus for improving the performance of WSD for Amharic texts.

This research intends to answer the following research questions in relation to WSD using part-of-speech tagged data for Amharic:-

- What is the performance impact of using part-of-speech tagged data for WSD?
- What is the optimal window size for the experiment while using part-of-speech tagged data for Amharic WSD?

1.3 Objective of the study

1.3.1 General Objective

- The general objective of this research is to investigate the application of part-of-speech tagged corpus to improve performance of Word Sense Disambiguation of Amharic text.

1.3.2 Specific Objectives

To meet the above general objective this study attempts to address the following specific objectives:-

- To study ambiguities in Amharic so as to understand word sense ambiguity problem in the language
- To study machine learning algorithms to be used in the experiment of WSD that uses POS information
- To use part-of-speech tagged data set for building WSD model
- To design the architecture of the WSD prototype that uses POS tag information
- To build and train WSD model using the POS tagged corpus
- To analyze and evaluate the performance of the model
- Forward conclusion and recommendations

1.4 Scope and Limitation of the Study

Due to lack of linguistic resource for WSD research in Amharic, the study was confined to 1031 Amharic sentences involving the different senses of the 5 polysemous words. Even if ambiguity can exist in a word, clause and sentence level; this study has concentrated on word level ambiguity due to its necessity to other levels as well as the indispensability of words for natural language processing systems.

In addition, the researcher has focused on lexical disambiguation unlike other types of ambiguities prevailed in natural languages like (Structural, referential, phonological etc).

1.5 Significance of the Study

This research work is conducted in order to investigate performance of WSD for polysemous words in Amharic after incorporation of POS information to the corpus. In addition to the beginning research works, this work seeks to add value to the previous works.

Therefore, the result of the study can be used to exercise developing an accurate WSD model since it can boost the performance of the variety of language applications (like MT, text retrieval, document classification, document retrieval etc.). Moreover, it will create opportunities for wider development of HLT which could be built for Amharic.

1.6 Organization of the Thesis

This thesis is structured into six chapters. The first chapter is an introductory part, which discusses the problem area leading to this research project, the general and specific objectives to be achieved in the research, the methodology to be followed and significance of the study.

The second chapter mainly revolves around literature review to discuss WSD using different approaches and ways of using corpus tagged with part-of-speech information for WSD prototype development.

The third chapter mainly revolves around the under-resourced language used for the study, Amharic, its writing system and the ambiguities in the language.

The fourth chapter is devoted to system architecture and methodology of the study. Hence, design of the proposed system architecture, discussion about methodology for data collection and the techniques of evaluation of the model were presented.

The fifth chapter is devoted to experimentation and discussion of the findings of the study. The last chapter, chapter six, is devoted to the final conclusion and recommendation based on the findings drawn from the study.

CHAPTER TWO

2 LITERATURE REVIEW

The purpose of this chapter is to provide a brief review of WSD. This is achieved by doing exhaustive literature review in the area of WSD to investigate WSD approaches, techniques and tools that were employed during the evolution of WSD researches since its inception. The comprehensive coverage of existing approaches is considered indispensible since it serves for understanding the central problems in WSD. The chapter also encompasses brief background and history, application areas and discussions on the three classes of machine learning methods that have been employed for WSD research.

2.1 Historical Background to Word Sense Disambiguation

WSD as an automated process of recognizing the word senses in context has got increased interest in recent years due to a wider need to enhance the language processing tasks. Even though the endeavor of improving the performance of WSD has increased in recent years its need has been detected during the early NLP applications like MT [35].

WSD has been formulated as a distinct computational task during the early days of MT during the late 1940s. The problem of WSD has been introduced by Weaver in 1949 with his work on MT. Automatic sense disambiguation approaches lacks world knowledge but it has been a focus of attention since the earliest times where computers were used for treatment of a language in 1950s. Until WSD has been acknowledged as difficult and hindering problem for the development of MT; researchers were considering the possible features which could enable to manage the difficulties for addressing the issues of WSD, of which considering context of target word, statistical consideration of the information of the words and senses, knowledge resources etc. were some to mention[36][90]. According to [37], one of the reasons for the difficulty of WSD is the vague nature of polysemous words. The other main reason which makes WSD such a difficult problem is the necessity of different knowledge or information sources.

The problem of WSD has been described as AI-complete problem in a sense that it can be solved only if all the difficult problems in AI (representation of common sense and encyclopedic

knowledge) are solved. The inherent difficulty of WSD as a hard problem got attention in Bar-Hillel's treaties on MT which asserted no means by which sense of an ambiguous word could be determined automatically [37].

At about the same time, a considerable attention has been given for representation of world knowledge particularly due to the emergence of semantic networks. The effort of addressing the problems of WSD has continued in the next two decades in the framework of natural language understanding researches as well as in the fields of content analysis, stylistic and literary analysis and IR.

According to Robert Navgali, the effort of addressing the problem of WSD has also continued in 1970s using AI approaches of language understanding [38], though was found out to be difficult due to lack of large amount of machine readable knowledge. Despite the above efforts, since WSD were largely rule based and hand-coded, it was prone to knowledge acquisition bottleneck. It has also been described in the work of Robert Navgali that the work has continued even during the time of large availability of machine-readable knowledge which have brought turning point during 1980s [39].

During 1980s, the availability of large-scale lexical resources such as OALD were drive for the replacement of hand-coding with knowledge automatically extracted from lexical resources but the disambiguation was still knowledge-based or dictionary-based.

In the 1990s, the statistical revolution swept through computational linguistics and WSD became a paradigm problem on which to apply supervised machine learning techniques.

During 2000s supervised techniques were best in accuracy as a result attention has shifted to coarser-grained senses, domain adaptation, semi-supervised and unsupervised corpus-based systems, combinations of different methods, and the return of knowledge-based systems via graph-based methods. Further works have continued since that time with the use of statistical methods and periodic evaluation of WSD systems [40].

2.2. Word Sense Disambiguation

Even if the study of automatic WSD dates back 1950, accurate disambiguation have only been achieved for many years to the text of tightly focused subject areas [140].

The task of designing WSD system so far has been mainly based on set of manually created rules (general context rules and template rules) for sense selection and was effective for doing WSD. The disambiguation tasks using rule based approach was challenging since it cannot be extended and needs much human effort initially. Hence, the research work on WSD has moved away from manually created rules towards automatically generated rules based on disambiguation evidence derived from the existing corpora available in machine readable form since the mid-1980s[91][42].

During the mid-1980's, the available corpora in machine readable form was able to provide textual definitions from a dictionary to provide evidence for the disambiguator. Lesk's work has demonstrated the use of existing corpora as sense disambiguation evidence to raise the possibility of building a disambiguator which can resolve the senses of many words though the accuracy of disambiguation with only limited testing was reported between 50-70% which actually was not high in comparison with later disambiguators. Lesk's work has paved way for other researches in the area using many different corpora types and sense selection methods [43].

Yarowsky [44] has applied semi-supervised learning on WSD by using the clues which could be obtained based on the nature of ambiguous words. The clues used by during the research work was sense per discourse, sense of ambiguous words which result similar clues to be seen on any given document, and the other clue is a sense per collocation of ambiguous word and consistent expression of the nearby content words for the disambiguation. In addition, it has been addressed that an accuracy more than 96% have been achieved on English ambiguous words with the incorporation of decision list algorithm and one sense per collocation approach has also been explained to be the one commonly exploited in an iterative bootstrapping procedure[44].

Thus, Getahun [32] has used sense per collocation concept while building Amharic WSD model. During the model development the researcher has used 10 collocations to the left and right of the target word (10 word window sizes).

2.3 Application areas of WSD

WSD is vital for many NLP applications and has been considered as an intermediary step for many of natural language applications rather than being used as an end by itself. Applications which directly or indirectly rely on WSD are SA, MT, NER, SP, IE, QA systems, IR, text mining, Text summarization, TE, Semantic Role Labeling, and CLIR etc. Some of the applications of WSD systems are briefed below.

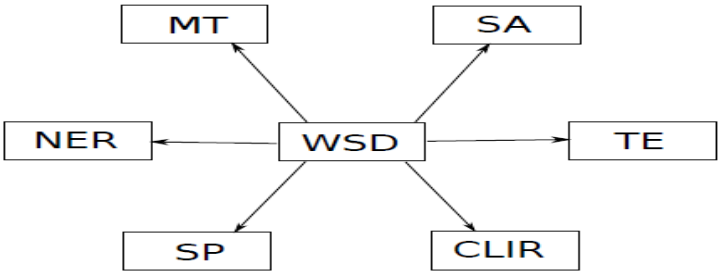


Figure-2.1 WSD as a heart for many NLP applications [17].

Machine Translation: - is an NLP application which has served as a drive to WSD system development [36]. Recently, some researchers investigated the importance of WSD systems in enhancing the performance of MT of which one is [47]. According to Marine Carpuat [47], WSD was found to boost the performance of statistical MT on fully phrasal multi-word disambiguation task applied on translation from Chinese to English. In addition, it was also experimented that WSD simplifies understanding of source language and generation of sentences in target language. However, MT donot use WSD system but it uses pre-determined lexicon for a given domain, handcrafted rules and statistical translation models. Hence, MT is also considered as the original and the most obvious application of WSD [21].

Information Retrieval: - is one of the NLP applications which benefits from WSD. An IR system which involves WSD as one component will have a capability to resolve ambiguities in some queries and hence it benefits a lot due to a reduction (elimination) of irrelevant hits. IR systems

which do not involve WSD as one component may retrieve many irrelevant documents for the queries involving ambiguous words so such systems benefit a lot by integrating WSD as one component [48]. According to Sanderson [49], early experiments have shown that at least 90% disambiguation is required by reliable IR systems in order an explicit WSD system to benefit. WSD were also considered indispensible for improving cross-lingual IR and document classification [21].

Speech processing: - is an NLP application which requires disambiguation for correct phonetization of words in speech synthesis. It is also used in segmentation and homophone (words which are spelt differently but pronounced in the same way) discrimination in speech recognition processing homophonous words [50] [51].

Due to the necessity of WSD systems to be used as a complement by variety of language applications, many international research groups are working on the area using different approaches. Despite the efforts done so far, no large-scale, broad coverage and accurate WSD systems have been built till today. Therefore, due to the importance of having WSD systems with improved accuracy the area is considered as an open problem in NLP for researchers till recently [52].

2.4 Approaches to WSD

The overall goal of WSD systems is to attribute the correct sense of polysemous word in a given context (discourse) for successful disambiguation tasks. Basically, there are two approaches to tackle the problem of WSD in NLP namely shallow and deep approaches [12]. According to Preeti [12], Shallow approach focuses on considering the surrounding words rather than understanding the text. In this approach, rules of word sense are extracted automatically from a training corpus of words tagged with their word senses [13]. Even if this approach is not considered powerful theoretically as deep approaches; it has been giving superior results in practice due to the computer's limited world knowledge.

In the case of deep approach, access to a comprehensive world knowledge will be performed to determine in which sense a word is used though this approach is not useable practically due to lack of MRD outside limited domains and for under-resourced languages in particular. Deep

approaches would be much more accurate than shallow approaches if such body of knowledge exists in every domain (computer representation of world knowledge)[14].

According to Ide and Veronis [15], there are four conventional approaches to WSD. These are, Knowledge based or dictionary-based, supervised, semi-supervised and unsupervised approaches as explained in greater detail below. All these approaches work by defining a window of n words around the target word.

2.4.1 Knowledge based Approaches

Knowledge Based approach involve external knowledge resources which define explicit lexicon (sense distinctions) for exploiting and assigning the correct sense of a polysemous word in context. This approach relies on knowledge resources like wordnet, thesauri, dictionaries, ontologies, collocations etc [53][54]. Systems which involve this approach take over the task of disambiguation by matching context with the information from prescribed knowledge source. During 1979 and 1980s, WSD experimental works have been done using initial knowledge based approaches on extremely limited domains though grading up of those initial works was hard. The main hindrance for use of knowledge based approach for end to end applications was lack of large-scale computational resources for evaluation, comparison and exploitation with feasible costs [55].

2.4.2 Corpus based Approaches

Corpus Based Approaches involve machine-learning techniques (supervised, unsupervised and bootstrapping techniques) to induce models of word usage from large collections of text examples. In this approach, the correct sense of a polysemous word will be figured out using context, the environment in which the word is used. Systems which involve this approach usually represent linguistic information for the context of each sentence (e.g usage of a polysemous word) in the form of feature vectors. The features are of part-of-speech labels, keywords, grammatical relationships, word collocations, topic and domain information etc.

2.4.3 Hybrid Approaches

Hybrid approaches involve aspects of both knowledge based and corpus based approaches in combination with the endeavor of attaining improved accuracy.

2.5 Machine Learning

These days, wider use of Internet and generation of sheer volume of data in different areas is becoming a common phenomenon. As a result, we are entering to the era of big data and such huge volume of data drives for an automated method of data analysis [56].

Learning is defined as a way of improving once performance with the use of past experience on a given task [57]. During the last decade a paradigm shift has been observed from rule based approaches to statistical and machine learning approaches in NLP research community. The paradigm shift has been observed in wider NLP applications like MT, preprocessing tasks involved in POS Tagging and WSD. As a result of which, practical tools like Machine Learning toolkits have been developed for use in the NLP experiments since then [58].

Machine learning is defined as a science of building systems that can learn from data. ML as a data-driven approach to problem solving is concerned with detecting patterns and trends that reside within the data for better decision making. ML has now become an important process useable in the modern data-driven economy. To carry out the pattern extraction and prediction task a set of complicated algorithms are incorporated to ML as its core component. The algorithms used in ML are considered as vital components developed from the diverse set of disciplines like statistics, computer science, robotics, computer vision, physics and applied mathematics during the past few decades [59].

In comparable to humans, who incorporate data in the form of experience, ML involve past data in the form of set of numerical and categorical attributes to train automated systems for later use to make accurate decisions when new events occur in the future. Thus, ML has a wide variety of application areas and its indispensability is likely to grow as more and more areas turn to it as a way of dealing with the available massive amount of data [56]. If we take 200 Amharic sentences containing the Amharic ambiguous word ‘derese’ (ደረሰ) as a set of initial training data for training the machine learning algorithm then the machine learning algorithm can be used to disambiguate the unseen context of the word ‘derese’ (ደረሰ) contained in other sentences.

Thus, ML can learn optimal decisions directly from the data which makes it accurate, automated, fast, customizable and scalable for optimized decisions unlike the rule based approaches. ML approaches are dependent on a tagged or untagged corpus evidence for training the model using supervised, semi-supervised and unsupervised algorithms [20].

Many ML methods have been proposed by researchers to deal with the problem of WSD. Of which, [60] has proposed use of supervised ML techniques, [44] proposed Semi-supervised ML techniques and [61] has proposed unsupervised ML techniques to be used for such experiments. Among the proposed ML methods, Supervised ML method has been very successful though the cost required for hand-labeling have hindered its wider applicability. unsupervised ML technique on the other hand though it hardly need sense tagging, the sense clustering results cannot be directly used in many NLP tasks as it lack sense tag for instances in the clusters. Unlike supervised and unsupervised ML methods, semi-supervised ML have become the currently experimented technique due to its incorporation of small sense tagged data together with unlabeled data, which is accessible in some richer extent [62].

2.5.1 Supervised Learning

Supervised ML is the use of algorithms that reason from externally supplied instances (training set) to form classes to differentiate new data. The goal of supervised ML is to build a model of the distribution of class labels in terms of predictor features. In order supervised ML to build the model it involves training and testing phases. During the training phase a sense-annotated training corpus is required, from which syntactic and semantic features are extracted to build a classifier using machine learning techniques and in testing phase the classifier tries to find out the appropriate sense for the word based on surrounding words present in the sentence. Supervised ML methods used for WSD problems are probabilistic based, similarity based, discriminating rule based and linear classification based. Probabilistic methods involves estimate of set of parameter such as conditional or joint probability distribution and context where the parameter is then used for assigning category to new sample which maximizes the conditional probability. In similarity based methods the categorization is done by comparing the features of new sample with the features of trained sample and assign sense of most similar

pattern (features). In case of methods involving discriminating rules, one or more selected rules associated with each word sense are involved to classify new sample and then assignment of sense will be followed based on their predictions [1].

Defining of word senses using supervised ML approaches equals construction of a semantic lexicon and manual annotation of word senses in text which is difficult and slow process [4]. Supervised WSD approaches yield very high accuracy at the expense of the costly resources required to be incurred in terms of time and manual efforts for sense tagging. To show how much supervised approach is a slow process [63], estimated that a longer time of 80 man-years of work is required for the semantic annotation of a corpus of 20, 000 ambiguous words for achieving high accuracy using the supervised techniques. Despite the higher accuracy for these approaches, the impracticability of creating corpora for all languages in all domains becomes a hindrance. In addition, ensemble methods which function by combining the different classification algorithms were also seen to yield better performance scores than using the classification algorithms separately [92].

The commonly used supervised ML algorithms are Naïve Bayes, Decision Lists, Decision Trees, Neural Network, SVM, and Exemplar-based [1] and are briefed as follows:

Naïve Bayes Method

Naïve Bayes method is a supervised ML method which uses probabilistic approach for estimating a set of probabilistic parameters that express the conditional or joint probability distributions of categories and contexts. Naïve Bayes classifier is based on Bayes theorem and hence the conditional probability is calculated for each sense of a word over the features defined (x1, x2,….., xm).

$$\arg\max_k P(k \mid x_1, \ldots, x_m) = \arg\max_k \frac{P(x_1, \ldots, x_m \mid k)P(k)}{P(x_1, \ldots, x_m)} = \arg\max_k P(k)\prod_{i=1}^{m} P(x_i \mid k) . \quad (2.1)$$

The probabilistic parameters of the model, P(k) and P(xi/k), can be calculated from the training set using relative frequency counts [1].

Decision Tree Method

Decision Tree algorithm is prediction based that uses knowledge source like sense tagged corpus for building the decision tree using which training is to be done. The yes-no classification rules are used to recursively partition the training data. Each internal node in the decision tree represents a feature value and each leaf node represents sense. After training phase got completed, the word to be disambiguated along with feature vector is traversed through the tree to reach leaf node where the sense contained in the leaf node will be considered for the word [1].

2.5.2 Unsupervised Learning

Unsupervised ML is an independent process where no supervision is involved during the learning step. Unsupervised corpus based methods are knowledge-lean and do not rely on external knowledge sources such as MRD, concept hierarchies and sense tagged texts. Unsupervised ML approaches are mainly clustering approaches where words and contexts are clustered. During clustering, each cluster corresponds to a sense of a target word. The goal of clustering is to group together elements in a way which maximizes similarity between elements in one cluster and to minimize similarity between elements belonging to different clusters. During unsupervised ML technique feature vector representation of unlabeled instances are taken as input and are then grouped into clusters according to a similarity metric [17]. These clusters are then labeled by hand with known word sensed though the main disadvantage being senses are not well defined [64]. Unsupervised approaches have the potential advantage of overcoming knowledge acquisition bottleneck and have achieved good results [65].

According to [66], clustering algorithms are divided in to two general categories as Hierarchical and partitioning. Partitional clustering algorithms includes K-means, Bisecting K-means, K-

Medoids etc. and those of Hierarchical algorithms are also subdivided into divisive and agglomerative.

Hierarchical algorithms are clustering algorithms concerned with creation of a nested partitioning of the data elements through an iterative merging or splitting. The final step of hierarchical algorithms is a tree structure (tree of clusters) called dendrogram. The dendrogram is concerned with provision of a visualization of how the algorithms form the clusters from the given data set. Hierarchical clustering algorithms are rigid in their decisions in that they do not open door to reconsider some wrong decisions which result bad clusters. A simple dendrogram which can clarify hierarchical algorithm is illustrated in the figure below.

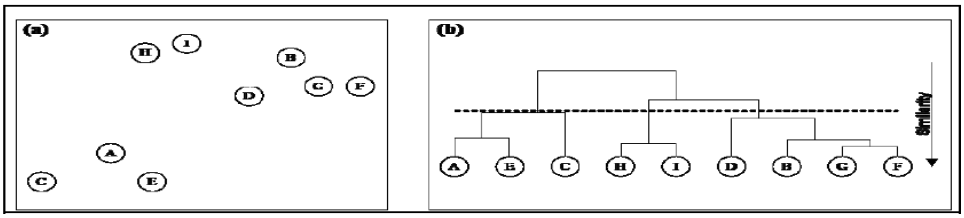


Figure-2.2 Dendrogram for hierarchical clustering adopted from [31].

Hierarchical algorithms are further subdivided as agglomerative and divisive based on how the hierarchical decomposition is formed. Agglomerative algorithms follow a bottom-up approach to form an all-encompassing cluster from the given separate data elements which were initially their own separate clusters. In order to form a single all-encompassing cluster agglomerative algorithms iteratively merge the data elements using similarity metric. Moreover, agglomerative algorithms are the commonly used clustering algorithms as compared to divisive and involve different versions which run with time complexity of $O(n^2 \log^n)$. The commonly used versions of agglomerative algorithms are: single-link, complete-link and average-link clustering [66].

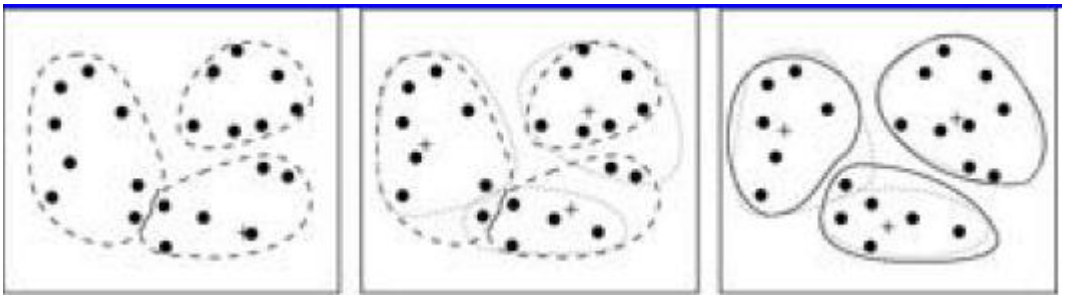
Divisive algorithms on the other hand follow a top-down approach to form separate clusters using an iterative splitting heuristics starting from a single cluster. This clustering algorithm has a worst time complexity of $O(n^2 \log^n)$ while splitting the given cluster into smaller clusters by taking all the $2^{(2n-1)} - 1$ possible clusters into consideration while a cluster is split at each step [66].

The other sub-category of clustering algorithms which often involve efficient running time is partitional algorithms. Unlike hierarchical algorithms which are concerned with creation of nested clusters, Partitional algorithms generate single partitioning cluster of predefined k-partitions from the given data set of size n where $k \leq n$, with the incorporation of optimization criterion. The basic partitioning methods typically adopt exclusive cluster separation where each object is supposed to be included in exactly one group. Most partitioning methods are distance-based where an iterative relocation technique is employed to improve the partitioning by moving objects from one group to another from an initial partitioning constructed at the start of partitioning. There are a variety of methods of measuring the quality of partitioning methods of which the commonly used one is objects in the same cluster are supposed to be as close as possible and objects in different clusters are far apart(i.e to attain high intra-cluster similarity and to attain low inter-cluster similarity). Partitioning methods use centroid to represent cluster center and are effective for small to medium data sets. There are different partitioning methods to mention some K-means, Bisecting K-means, K-medoids etc [66].

K-means

K-means algorithm is one of the simplest and commonly employed families of partitioning algorithms. It uses a concept of centroid, center of a cluster, calculated as the mean value of the points in the cluster. Centroid of a cluster is pseudo-element that represent center of all elements in a given cluster. The algorithm firstly selects k-centroids of the clusters among the data set and computes the Euclidean distance between the remaining objects and the already selected centroids. The algorithm assigns the unassigned objects to the closest cluster-mean based on the already calculated Euclidean distance. K-means iteratively maximize inter-cluster similarity by computing centroid for the objects in each cluster assigned in the previous iterations. The iteration will continue again and again by reassigning the objects in the clusters to the newly computed centroids until the assignment reaches to the stage where no observable change is brought relative to the formerly done iterations. The time complexity of this algorithm is $O(nkt)$, where n is the total number of objects, k is the number of clusters, and t is the number of iterations and hence it is relatively scalable and efficient in processing large

data sets. Despite the benefits, k-means algorithm is susceptible to be affected by outliers since they have the tendency to distort the mean value of the cluster [66].



a. Initial Clustering b. Iterate c. Final Clustering

Figure-2.3 clustering of a set of objects using K-means adopted from [66].

K-medoid

K-medoid algorithm like k-means is a family of partitional clustering algorithms designed to alleviate the problem of susceptibility of k-means algorithm in the presence of outliers. K-medoid algorithm is similar to that of k-means except in the usage of mean to represent centroid of cluster in the case of k-means. Initially, K-medoid algorithm randomly assigns k elements for the k-clusters then during each iteration other representative object will replace the existing when it is found out to have improved squared-error criterion. When the running time complexity of k-medoid is taken in to consideration, it is $O\ (k(n - k)^2)$ and as it could be inferred when the value of n and k is large it becomes costly even from k-means [66].

2.5.3 Semi-supervised Learning

Semi-supervised ML techniques involve training information like in supervised but the information given at initial training phase is less. Here only critic information is available, not the exact information. Semi-supervised or minimally supervised methods are gaining popularity because of their ability to get by with only a small amount of annotated reference data while often outperforming totally unsupervised methods on large data sets. There are a host of

diverse methods and approaches, which learn important characteristics from auxiliary data and cluster or annotated data using acquired information [64].

Semi-supervised learning framework involves seed examples which is a set i of independently identically distributed examples $x1,\ldots\ldots,_{i} \in X$ together with the corresponding labels $y1,\ldots,y_{i} \in Y$ will be given like that of supervised learning framework. In addition to the labeled examples, it also involves unlabeled examples of $x_{i}+1,\ldots,x_{i+u}$. Thus semi-supervised learning use combined information of labeled and unlabeled examples for achieving better classification performance that could be obtained by discarding the unlabeled data and doing the supervised or unsupervised by discarding the labels. Thus, Semi-supervised learning got greater theoretic and practical interest due to the higher accuracy at a reasonable annotation cost [62].

Enormous researches have been carried out to devise semi-supervised learning technique which functions by involving both labeled and unlabeled data. Many of the researches based on the semi-supervised learning techniques have considered data (labeled and unlabeled) from the same distribution but [66], have researched different techniques with the incorporation of varied types and amounts of labeled and unlabeled data from different distributions. The researchers have carried out an empirical study on various semi-supervised techniques and showed the effect of size of labeled and unlabeled data sets, sample selection bias on semi-supervised techniques, effect of independence or relevance amongst the features etc.

Semi-supervised machine learning algorithms involve both classification and clustering assumptions while employed in WSD. In one sense these algorithms are used when supervised learning is fed with more and more unlabeled data for the purpose of classification of the unlabeled data to a specific class and in the other case they are used when an unsupervised algorithm is provided with more and more labeled data for the sake of clustering. So, the two basic assumptions i.e the Manifold and Cluster assumptions incorporated in semi-supervised learning will be discussed below according to [32].

Semi-supervised classification (Manifold assumption): involves both labeled and unlabeled data within the training data. During this assumption, the already labeled seed data exploits the unlabeled data for building the training data set. This algorithm performs classification like that

of supervised machine learning algorithms but incorporation of unlabeled data in training data set creates a difference since supervised machine learning algorithms only involve labeled data in training data set. The data which is captured is represented as a graph where the seed data correspond to the vertices and the pair-wise similarities as edge-weight in the graph. According to this assumption, data with similar inputs are supposed to have similar categories/class [68][44].

Unlike supervised learning which uses fully labeled training sample, semi-supervised learning involves two distinct possible goals to be met as it involves both labeled and unlabeled data in the training data. The two goals are inductive and transductive learning as of [91] as briefed below:

Inductive semi-supervised learning: involves learning a function f from the training sample $\{(xi,yi)\}^{l}i=1$, $\{xj\}^{l+u}$ $j=l+1$ where f : X -> Y is expected to be a good predictor on future data beyond $\{xj\}^{l+u}$ $j=l+1$. This learning algorithm is used for predicting labels on future data and it can be considered as an analogy to in-class exam where students are expected to get prepared for all possible questions which appear in the exam. What makes inductive semi-supervised learning similar to that of supervised learning is on the usage of separate test sample to estimate the performance using the data which is outside of the training data.

Transductive semi-supervised learning: involves a simpler function f for prediction of labels for the unlabeled instances on the training samples {(xi,yi) li=1,(xj)l+u j=l+1}. Transductive semi-supervised learning trains the training function f: xl+u -> yl+u, the function is not supposed to predict class labels for the instances outside the training data. Transductive semi-supervised learning is like an exam comprising of questions which has already be given as home-take and students expect only questions selected from their home take exam.

Semi-supervised clustering (Cluster Assumption): involves training data that encompasses large amount of unlabeled data together with some seed (labeled data). Basically the task of this semi-supervised machine learning is clustering like unsupervised machine learning algorithms but with enhanced performance of clustering due to the incorporation of labeled data in the training data set. The clustering result will be consumed for better classification. Co-

training, self-training (ADtree, Adaboost, bagging, semiboost) and semi-supervised SVM such as SMO algorithms are incorporated in this assumption of clustering [68] [69].

Of the two assumptions underlying semi-supervised machine learning algorithms, the researcher prefers to use the clustering assumption which has been used by [32]. The main reason why the clustering assumption is preferred in this research work is it will be easy to compare this research output with [32]'s work if the algorithms used are the same. Hence, co-training and bootstrapping (self-training) will be discussed together with the algorithms used like ADboost, bagging, semiboost and ADtree below.

Bootstrapping

Bootstrapping works based on Yarowsky's supervised algorithm that use Decision lists and is devised by [140]. One of the first bootstrapping algorithms used in computational linguistics is that of [140] which was applied in WSD. Yarkowsky's algorithm incorporates set of labeled examples called seeds and unlabeled examples which comprise 90 % of the training data. The algorithms perform iteratively in which a decision list learner is built with the corpus of seeds and applied to the unlabeled data. In the subsequent iterations the algorithm gets rules of the seeds plus those of best confidence acquired from the unlabeled set and a new learner will be built and the process will continue until reaching some training parameters. A couple of strong assumptions have been made by the algorithm regarding the language which makes the model building phase to become quite small compared to the supervised analogue of this algorithm as briefed below:

One sense per collocation- is an assumption of the strong dependency of the sense of a word on neighboring words.

One sense per Discourse – is the other assumption which deals the high portability of a single sense of a word contained in every document.

The algorithm uses the above assumptions for identifying a set of seed words which can act as disambiguating words as an initial task. Secondly, a decision list will be build based on the seed data after that the entire sample set will be classified using the decision list generated previously. Lastly, using the decision list as many new words as possible are classified in order

to identify their senses and using these words along with their identified senses new seed data will be generated with repeated steps until the output converges to a threshold value. The bootstrapping algorithms which are used in this research will be discussed below.

Co-training

Co-training is an alternative bootstrapping algorithm which has been applied since 1998 [70]. Co-training involves two views (a learning classifier) on the set of mutually exclusive features which are applied one by one alternatively on each learning iteration.

Bagging

Bagging is a bootstrap aggregation which is applied on a training set (Di) which are sampled with replacement from the original set of tuples (D) for each iteration. This algorithm works with a majority vote where each candidate assigned for the selection have a chance to be selected many times and it also involves chance not to be selected even once since the selection of the samples is carried out with replacement. The classifier model will be build using the training set to predict the class of unknown tuple X. The class label of the unknown tuple is assigned based on the majority vote. The bagging classifier involves greater accuracy due to the composite model involved for reducing the variance of the individual classifiers at each iteration unlike a single classifier derived from the original data.

AdaBoost

AdaBoost(short for Adaptive Boosting) is a boosting algorithm which is mostly applied when boosting the accuracy of a learning method is considered important. Adaboost works by initially assigning an equal weight of 1/k to each member of the training tuple while acting on an initially given class-labeled tuples. The algorithm involves rounds to generate n classifiers for the ensemble in n rounds. During generation of classifiers in any round I using tuples from an initially given training set it involves sampling with replacement though the selection is based on its weight. The weights of the training tuples are adjusted according to how they were classified after a classifier model is derived from the training tuples [66].

SemiBoost

Pavan [68] has described that Semiboost algorithm is built by assigning labels to the unlabeled samples on the basis of two main criterion as described below:

- Points in a cluster among the unlabeled samples must share the same label
- Those unlabeled samples which are highly similar to the labeled samples must share its label.

Thus in each iteration, a classification model will be learned using the algorithm while being applied on a training data set. The models learned at each iteration will be combined linearly to form the final classification model and applied to form both semi-supervised classification and clustering.

ADtree

According to [71], ADtree algorithm is an algorithm which requires less training and computational cost unlike that of AdaBoost. This algorithm is widely used in real-world applications even to a greater extent from AdaBoost due to its strength in handling over fitting problems.

2.6 WSD related works for Amharic language

To the best of the researcher's knowledge some of the WSD works done for Amharic were:

Wube [28] has designed and developed a rule based parser with the intent of resolving structural ambiguity for Amharic language. During the study, the Wube [28] has used four hundred sentences, which involve fifty ambiguous words, randomly selected from different books to develop the parser. In addition, the researcher has used bottom-up as parsing strategy to construct the parses and depth first search strategy for invoking the rules and achieved performance score of 86% despite the small size Amharic texts used for rule induction. The performance measure has been obtained by comparing the automatically parsed sentences with those manually parsed.

Teshome [29] has researched on WSD using semantic vector with a goal of improving the accuracy of IR system for Amharic legal texts collected from Ethiopian postal code articles.

During the research, the researcher has developed an algorithm based on distributional hypothesis stating that words with similar meanings tend to occur in similar contexts. During the task of disambiguation, the researcher computed context vector of each occurrence of the words where context vector is derived from the sum of the thesaurus. In addition, the researcher has constructed the context vector by associating each word with its nearest neighbors and vectors of the context words. Lastly, for the sake of evaluation of WSD system the researcher used artificial (pseudo) words unlike real words due to the cost required for sense annotation and the algorithm has been mentioned to be superior over Lucene's one.

The third work on the area has been carried out by Solomon [30], where the researcher used supervised machine learning approach by applying Naïve Bayes algorithm. For the experiment, a total of 1045 Amharic sense example sentences were collected from BNC. Five ambiguous Amharic words '**አጠና**' (eTena), '**መሳል**' (mesal), '**መሣሣት**' (me`sa`sat), '**መጥራት**' (metrat), and '**ቀረጸ**' (qereSe) were used in the experiment. To collect the sense example sentences firstly Amharic-English dictionary has been used to get sense of each ambiguous words which then was converted to English and being used for retrieving sentences containing that sense which then after require transliteration back to Amharic. Lastly, the researcher has carried out fully labeling of the dataset to use it for the classification algorithm, Naïve Bayes. Moreover, the researcher has also experimented and determined an average optimal window size of three-three to be enough for Amharic WSD task. The researcher has used Weka 3.6.2 tool and achieved accuracy in the range of (70-83)% for developed prototype model.

The fourth attempt has been done by Solomon [31]. The research was focused on experimenting unsupervised machine learning approach for WSD to selected polysemous Amharic words already used by [30]. After acquiring the 1415 sense example sentences preprocessing tasks were carried out to use it for the experiment. Five clustering algorithms (simple K means, EM, simple, average and complete link) have been used during the experiment. The researcher has used unsupervised machine learning paradigm in that it alleviate the knowledge acquisition bottleneck faced by supervised methods. Nevertheless, as fully unsupervised methods neither exploit any dictionary nor rely on a shared reference inventory of senses they end up with lesser accuracy while seen in comparison to that of

supervised methods. As of the experimental result, simple k means and EM clustering algorithms have better accuracy on the task of WSD using the corpus involving ambiguous words. The accuracies achieved in the experiment using simple k means and EM were in the range 65.1-79.4% and 67.9-96.9% respectively though worst accuracies were achieved using single and average link clustering.

The fifth research work on Amharic WSD was by Getahun [32], who has employed semi-supervised machine learning approach to build a model for disambiguation of Amharic text. The researcher has used semi-supervised machine learning paradigm to compromise the knowledge acquisition bottleneck of supervised approach and worst performance achieved by unsupervised approach. The researcher has carried out the research work using 1031 Amharic texts which involve 5 polysemous words. The experiment involves 2 clustering algorithms (simple K-means and EM) and 5 classification algorithms (adaboostM1, bagging, ADtree, SMO and Naïve Baye). The experimental result depicts performance score of 88.47% using ADtree, 87.40% using SMO and 83.94% using AdaboostM1 has been achieved during the development of prototype model using semi-supervised paradigm. In addition, the researcher has shown that window size of 2-2 and 3-3 is enough for developing WSD system for Amharic.

The other research work on Amharic WSD was by Hagerie [33], who has experimented to model WSD using Adaboost and Bagging ensemble classifiers involving five decision tree algorithms as base classifiers. The experiment has involved 1770 sense examples of eight ambiguous Amharic words: ‘ale’, ‘atena’, ‘bela’, ‘derese’, ‘ekebere’, ‘qerbe’, ‘melese’ and ‘atena’. The experiment on Adaboost and Bagging was carried out using five decision tree algorithms (DecisionStump, J48, RandomForest, RandomTree and REPtree) as base classifiers and the result depicted that ensemble learning algorithms outperform the base decision tree algorithms. In addition, it was reported in the experimental result that Random forest was the best base decision tree classifier in providing the possible performance among the rest decision tree classifiers for ensemble classifiers and both ensemble algorithms were seen to have comparable performance i.e. only a percent more performance was gained by Bagging over Adaboost. Moreover, the performance of ensemble classifiers was seen to have better performance of 79.70% for Adaboost and 80.46% for Bagging when the window size used was two.

Lastly, Birhane [34] has done WSD research on Amharic text using ensemble of Naïve Bayesian classifiers. During the study, 1415 texts which involve distinct senses of 6 polysemous words have been used. The polysemous words which were used were: '**አከበረ**' (ekebere), '**አለ**' (ale), '**አጠና**' (atena), '**በላ**' (bela), '**ደረሰ**' (derese) and '**ተነሳ**' (tenesa). The researcher has used six combination rules including Max-Rule, Min-Rule, Medium-Rule, Performance Based and Joint Product rule where the inputs for the combination rules was that of Naïve Bayes classifiers. During the experiment Joint Product Rule has achieved maximum improvement of (72.6 and 82.13%) for the Amharic words 'ale' and 'derese' respectively.

When the above mentioned previous works are seen in comparison, the first work is different from the rest of the works in that it was domain dependent and has used pseudo words during the experimentation. The second work using supervised approach has faced a problem of knowledge acquisition and the third word using unsupervised approach though inexpensive has end up with least accuracy. Ensemble classifiers experimented using Naïve Bayes and boosting algorithms (Adaboost and Bagging) has also resulted with less accuracy though better than that of unsupervised and comparable to supervised approach. Of all the above works for Amharic WSD, semi-supervised approach used by [32] has scored better accuracy.

Lastly, different literatures addressed the importance of linguistic information such as POS tag to add value to the current performance scores achieved during the development of WSD prototype systems using different languages. To the best of the researcher's knowledge no work has been done on the investigation of effect of POS tag information on the performance of WSD prototype model development. Hence, this work is different from the above works in that, the researcher has planned to use a corpus with part-of-speech tag information for the experiment involving the three classes of machine learning methods.

Summary

In this chapter the brief history of WSD has been covered since its inception during the 1940's and 1950s'. Since 1940s, different researches have been done using different approaches like knowledge-based and knowledge-poor (corpus based). Improvements have been seen but researches are still underway to develop better WSD systems due to its wider application areas

like text and speech processing systems, IR systems, MT etc. As it has been mentioned in this chapter, lack of standard linguistic resources have hindered experiments for Amharic despite initial works done so far using small size corpus collected by researchers. The study has been confined on the shallow approach unlike deep approach since experiments using deep approach requires standard linguistic resources like MRD, thesaurus etc. Lastly, the algorithms used in this study have been described after survey to different approaches and algorithms used on previous research works on the area.

Despite the efforts so far, the problem of polysemy is deemed open to researchers since the current state of the art accuracy is long-way far from the capability of human beings to overlook the problem.

CHAPTER THREE

3. THE AMHARIC LANGUAGE

3.1 Overview of Amharic Language

Amharic also known as Abyssinian, Amarinya, Amarigna, and Ethiopian is the working language of Ethiopia, a country having a population of over 90 million at present [72] [73]. Ethiopia is a linguistically diverse country with more than 80 languages used for the day-to-day communication. Amharic is believed to be derived from Geez, a liturgical language of Ethiopia since 4th century AD [74].

Amharic is the second most widely spoken Semitic language in the world next to Arabic. Of the diversified languages spoken in Ethiopia, Amharic is serving as a mother tongue by a large segment of the population in the northern and central regions of the country and as a second language by many other regions [73]. It is the most commonly learnt language in Ethiopia next to English and serve as the official and working language in the country [75].

Amharic has been used in public life, as a tool for communication in commercial activities, in specialized publications and in other spheres of daily life. In public life, it has been used in political discourse, civil service, courts and in the parliament. The language has also been used as an important commercial language where it has widely been used in marketing, in business transactions and banking. In addition, the language has long tradition in music and the creative arts, resulting in a rich heritage in this era. In the educational front, it is serving as a medium in primary and secondary school levels [72].

3.2 THE AMHARIC WRITING SYSTEM

The name of the script which has been given for writing purpose of Ethiopian and Eritrean languages is Ethiopic (“Fidel”). Ethiopic and Geez is name of the script used by foreign linguists and Amharic speakers in the country respectively. The script descends from ancient Semitic family and also belongs to Afro-Asiatic super-family [76].

According to [76], Amharic is one of the African languages with three writing systems called the Amharic syllabary, the Roman alphabet and Arabic script. The Amharic syllabary which is uniquely Ethiopian writing system has derived from the writing system of ancient south Arabian inscriptions. The syllabary which has similarity with some Semitic languages like Arabic in having vowel marks added to basically consonant letters is used for Ge'ez, Amharic and Tigrigna with slight modifications.

Ge'ez took its script from the ancient Arabian language mainly attested in inscriptions of the Sabean dialect. Ge'ez took 24 of the 29 original Sabean symbols with the incorporation of two new symbols to represent sounds of Greek and Latin loan words not found in Ge'ez when it became the spoken and written language in common use in northern Ethiopia. The writing system has also been modified from left to right in addition to the modifications added to most of the symbols [76].

When Amharic and other languages replaced Ge'ez both for spoken and written, Amharic took all the symbols in Ge'ez while later added some new characters such as ቸ, ጨ, ጀ, ኘ, ቨ, ሸ, ኸ, and ኧ for representing sounds found in Ge'ez [77].

The present writing system of Amharic, taken from Ge'ez, contains 34 consonants (which are base characters) and six other forms in a horizontal position called orders i.e. among the seven orders the first order is consonant but the other six orders are consonant tangled with vowel as it is shown in the figure below. The Amharic writing system is often called syllabic rather than alphabetic since the seven orders represent syllable combinations consisting of a consonant followed by a vowel though there is some opposition [77]. The 34 basic characters and their orders give 238 distinct symbols.

vowel	-	u	i	a	y	E	o
h	ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
l	ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ

Figure-3.1 Adapted from [32] work without modification

3.3 AMHARIC PUNCTUATION MARKS

Languages having their own writing system or script have punctuation marks which are used for different purposes like separation of words, as a pause between words, termination of sentences etc. and Amharic is no exception. The Amharic writing system also involves about 17 indigenous and loan punctuation marks for use in practical communications though only some are used in digitally written texts and have representations in Amharic software [78]. The commonly used punctuation marks in Amharic are: 'hulet neTb' (word separator) represented as a colon (:), 'arat neTb' (Sentence separator) represented as four square dots arranged in square pattern (: :), 'netela serez' (list separator) which is represented as (፣) followed by an ASCII space , 'derib sereze' which is represented as (፤) and use of question mark formerly "…" though replaced by an a question marked borrowed from English language(?) etc. In Amharic writing system, punctuation marks used for sentence termination are among (!,?, or ::) [79]. The following figure shows some of the commonly used punctuation marks in Amharic with their English equivalents.

Amharic	English
፡	White Space
።	.
፤	;
፣	,
፧	?

Figure-3.2 Most commonly used Amharic punctuation marks with their English eqivalents adopted from [78].

3.4 SYNTACTIC STRUCTURE OF AMHARIC

The syntactic structure of a sentence refers to the way words in a sentence are organized and related to each other. In addition, it indicates how the words are grouped together in to phrases, what words modify other words and used to pin point words which are of central importance in the sentence [80].

According to [31], like in any other languages, the organization of words in a sentence together with the syntax in Amharic define the syntactic structure for the language which is generally SOV. If we consider the Amharic declarative sentence which is an equivalent of the English sentence "He went to school."

"እሱ: ወደ: ቤተ-መጽሃፍት: መጣ::"

"**እሱ** (pronoun)" is the subject, "**ቤተ-መጽሃፍት**" is the object and "***መጣ***" is the verb.

Most of the time, the pronoun "**እሱ**" will be left since it could be implicitly understood from the verb "***መጣ***".

On the other hand if we consider the structure of an interrogative sentence in Amharic which is an English equivalent of "did he go to school?" is:

"**እሱ ወደ ትምህርት ቤት ሄደ**?" where the structure of the sentence does not show any change but the punctuation mark since it is an interrogative.

In some cases words which implicate the interrogative sense of the sentence will be added to the end of the sentence as seen in the example below:

"**እሱ ወደ ትም ህርት ቤት ሄደ**?"

3.5 Part of Speech Tag for Amharic

Natural language which is used by human beings as a means of exchanging ideas and for conveying complex thoughts involves system of rules and conventions like alphabets, word combinations, a pause, etc. The rules and conventions in the language determine the resulting information to be communicated. Amharic like any other natural languages is also governed by such rules. Moreover, there are features of a language like POS which are used to define a linguistic category of lexical items or words, which are the important building blocks of any language [79].

POS, which is also known as word class, lexical class or lexical category, is defined as a linguistic category of lexical items or words. It is defined by morphological and syntactic behavior of the words. The known part of speech include: nouns, verbs, adjectives, adverbs, prepositions, interjections etc. Word category of one language may be different from that of other language. Part of speech provides important information about a word and its neighbors in language processing.

POS tagging, which is one of the fundamental processes in the field of Natural Language processing (NLP), is an area of research for many languages and fundamental processing step in NLP and language automation activities, i.e., the capability of a computer to automatically tag a given sentence using an appropriate POS like adjective, adverb, noun, verb, pronoun etc.

During the process of POS tagging, the string which is used as a label is called a tag and the set of such labeling strings is named as tag set. The task of POS tagging can be done manually, while experts assign a tag to the words in text, and automatic when an NLP tool or program designed for POS tagging is used for carrying out such task[79][93].

POS tagging is challenging due to the nature of words to have more than one POS tag. Despite such challenging nature of a language, many POS tagger systems have been developed for many languages like English, French, Spanish and German etc and such POS taggers have also been developed for local languages in our country like for Amharic and Afan etc. [94].

3.6 AMBIGUITIES IN AMHARIC

Amharic as in any other human languages involves ambiguities since "It is usually assumed that natural languages are inherently ambiguous...." [81]. There are different sources for ambiguities in a language such as placement of a pause within structures', categorical diversity, Homonymy etc. There are six types of ambiguities that are involved for Amharic language as descried below as of [5].

3.6.1 Phonological Ambiguity

Phonological ambiguity is a type of ambiguity which is resulted due to the placement of a pause within the structure of the word when pronounced. Example of such type of ambiguity is:

Example-1 i. [ደግ + ሰው] ነበር (to mean he was a kind man)

deg + sew neber

ii. [ደግሰው] ነበር (to mean they have made a ceremony)

Degsew neber

Example-2 i. [ሥራ + ስሩ] ጥሩ ነዉ (to mean it is good to work)

Sira + siru tiru new

ii. [ሥራስሩ] ጥሩ ነዉ (to mean various roots are good)

sirasiru tiru new

In the above examples the placement of the plus sign within the structure of the word shows the pause in between [5].

3.6.2 Lexical Ambiguity

Lexical ambiguity is a type of ambiguity which is resulted due to the occurrence of two or more meanings of a word in a given context. It has been used to refer for a lexical unit belonging to different part-of-speech categories with different senses or to a lexical unit in which there is more than one sense but the different senses belong to the same part-of-speech category as of [82][32]. [5] has described categorical ambiguity, Homonymy and Homophonous affixes to be some examples for the possible caused of Lexical category which are briefly discussed below.

Categorical Ambiguity

Categorical ambiguity is a type of lexical ambiguity which is due to categorical diversity. Lexical elements which have identical phonological form but different word class are named as categorically ambiguous [5].

Examples: 1. **አክርማ ሰጠችኝ**

i. She gave me Akirma (a kind of grass). [With nominal meaning]

ii. She gave me something after delaying it for some time. [With verbal meaning]

2. **ስጋ ብላ ይሆናል**

i. Meat will become useless. [Body becomes useless]

ii. She might have said 'meat' to you.

Homonymy

Homonyms are lexical items which have the same phonological form but different meanings. Homonyms are considered the other sources of lexical ambiguity as described in the examples below [5].

Examples: 1. **ዓርብ ጠፋ**

i. He disappeared on Friday (on one of the week days).
ii. Weaver's frame is lost.

The above example become ambiguous in that "**ዓርብ**" is used to mean the day of the week or weaver's frame.

2. **በወሬ አልፈታም**

i. I will not be released in a month.

ii. I will not get frustrated by any rumor.

Homophonous Affixes

The other cause of lexical ambiguity is homophonous affixes where affixes server for the existence of different word classes. The examples below describe homophonous affixes [5].

Examples: 1. **ቤቱ ፈረሰ** (bet+u ferese)

i. The house is destroyed.
ii. His house is destroyed.

2. **ሰው አማህ**(sew amma + h)

i. you backbit someone.

ii. Someone backbite you.

3.6.3 Structural (syntactic) Ambiguity

Structural ambiguity is a type of ambiguity which is resulted due to the different possible positions of the syntactic constituents in a given context [31].

Example:

የአረብ ታሪክ አስተማሪ

The above example refers to two different meanings due to the orientation of the structure as described below.

i. In one sense it refers to mean "Ye-areb tarik estemari "(he is teaching history of Arab)

ii. In the other sense it refers to mean "The history teacher is from Arab family".

3.6.4 Referential Ambiguity

Referential ambiguity is a type of ambiguity which arises when a pronoun has more than one possible antecedent. There are two types of pronouns in Amharic i.e. free and bound, which require Antecedents with which they will be matched in number, gender and person. Referential ambiguity hence is due to the different readings as there are antecedents a seen in the example below [5][31].

Example-1 . **ካሳ ስለተመረቀ ተደሰተ**

i. Kassa was pleased because he graduated.

ii. Somebody was pleased because Kassa graduated

3.6.5 Semantic Ambiguity

According to [5], polysemic, idiomatic and metaphorical constituents are some of the possible causes of semantic ambiguity as briefed in the next examples.

Examples:

1. Semantic ambiguity due to polysemic constituent
   a. **መብራቱ ጠፋ**

The two different senses of the above polysemic constituent are described in the following two contexts.

   i. The light went off. (Light in the context of supportive for seeing things)
   ii. Mebratu(a person named Mebratu) disappeared.

2. Semantic ambiguity due to Idiom
   a. **በሬ ወለደ**

   The two different senses of the above idiom are:

   i. It refers to mean something “unheard of” or “something which is impossible to happen” (in the idiomatic sense).
   ii. Literally it refers to mean the ox gave birth (which is factual).

   b. **ቤቱን ሰው አያውቀውም**

   The two different senses of the idiom “**ቤቱን ሰው አያውቀውም** ” are:

   i. In the idiomatic sense it refers “He never invited anyone to his home”.
   ii. The literal meaning refers to “No one knows his home”.

3. Semantic ambiguity due to Metaphoric senses
   a. **አራስ ነብር**

   Like the idiomatic senses, semantic ambiguity which is due to metaphors involves literal and non-literal (metaphoric) senses.

   i. The literal meaning refers “leopard with new-born cubs”.
   ii. The non-literal (metaphoric) sense is ‘Irascible, hot tempered'.

   b. **የጀርባ አጥንት**

   i. Literally it is used to mean “backbone”.
   ii. Metaphorically it refers to “something which provides main support”.

3.6.6 Orthographic Ambiguity

Orthographic ambiguity is a type of ambiguity which is caused due to orthographic reasons as the system does not show distinctions between geminate and non-geminate sounds. The intended meaning of the word could be identified based on the context though as it could be seen in the following examples [5].

Example-1 **መኪናው ይሰራል** (mekinaw yiseral)

i. The car works.
ii. The car will be repaired.

CHAPTER FOUR

4. SYSTEM ARCHITECTURE AND METHODOLOGY OF THE STUDY

4.1 Data Collection

As it has been addressed in the previous chapters, in this experimental study; the researcher has used corpus based approach during the experiment. As the former researchers on WSD using Amharic have depicted; it is challenging to acquire sense annotated corpus for WSD studies due to lack of standard sense annotated corpus or context based repository (Wordnets) for the language [30][31][32]. Hence, the researcher believes it is worth to use a corpus which has been collected and preprocessed by [32].

As a corpus used for WSD prototype development is not supposed to be bound to a specific domain due to a problem it results in limiting sense of a word to one sense, Getahun [32] has collected the data from different sources like Walta Information Center (Amharic news), Addis Admas news, VOA news, Cyberzena new, Fana Websites and Amharic bible. The data set involves 1031 Amharic sentences involving two different senses for each of the five ambiguous words as seen in the table-4.1 below.

Ambiguous words	Senses	No. of senses	Total
Atena (አጠና)	Strengthen	111	215
	Study	104	
Derese (ደረሰ)	Reach	100	207
	Mature	107	
Tenesa (ተነሳ)	Stand	100	200
	Cause	100	
Ale (አለ)	Say	107	208
	Live/Present	101	
Bela (በላ)	Eat	100	201
	Speak	101	
Total			1031

Table-4.1- Amharic ambiguous words and their sense adopted from [32]

As seen in table-4.1 above, a balanced distribution of senses of the Ambiguous words has been involved for the experiment to get improved performance for WSD study. According to [83], performance degradation could be faced by machine learning algorithms when unbalanced distribution of the senses is used for training and testing samples.

The dataset comprised three sets of data each comprising of 1031 Amharic sentences involving the five ambiguous words. The first set of 1031 Amharic sentences were fully unlabeled to be used for the experiment using unsupervised machine learning algorithms, the second set of 1031 sentences were fully labeled to be used for the experiment using classification algorithms and the third set of 1031 sentences comprises of labeled and unlabeled data (i.e. among the 1031 sentences 14% are sense tagged). Lastly, as the corpus which was acquired from Getahun [32] was already preprocessed, the tasks which might have been faced preliminarily like selecting Amharic polysemous words, acquiring of sense examples for the selected polysemous words and preprocessing (tokenization, stop word removal, stemming, normalization and transliteration) has not been faced in this study.

4.2 POS Tagging of the Corpus

In this study, CRF POS Tagger which was developed by [84] has been used in order to attach POS tag to all the words available in the 1031 sentences used in this study. The POS tagger has been used to tag all of 3096 sentences.

4.3 Proposed System Architecture

In this section, the flow of activities to WSD experiment has been presented using the proposed architectures. Three architectures were proposed and used in this study since three of the machine learning paradigms were planned to be investigated using POS tagged corpus.

The first proposed architecture presented below in Figure-4.1, accepts 1031 fully labeled sentences involving two senses of each of the five ambiguous words. Next, the fully labeled dataset will be given to the Amharic POS tagger model developed using CRF++ tool [84] so that each word involved in the sentences is tagged with POS tag information. Lastly, fully labeled dataset with POS tag information was given to the classification algorithms to build the WSD prototype model and evaluated for its accuracy.

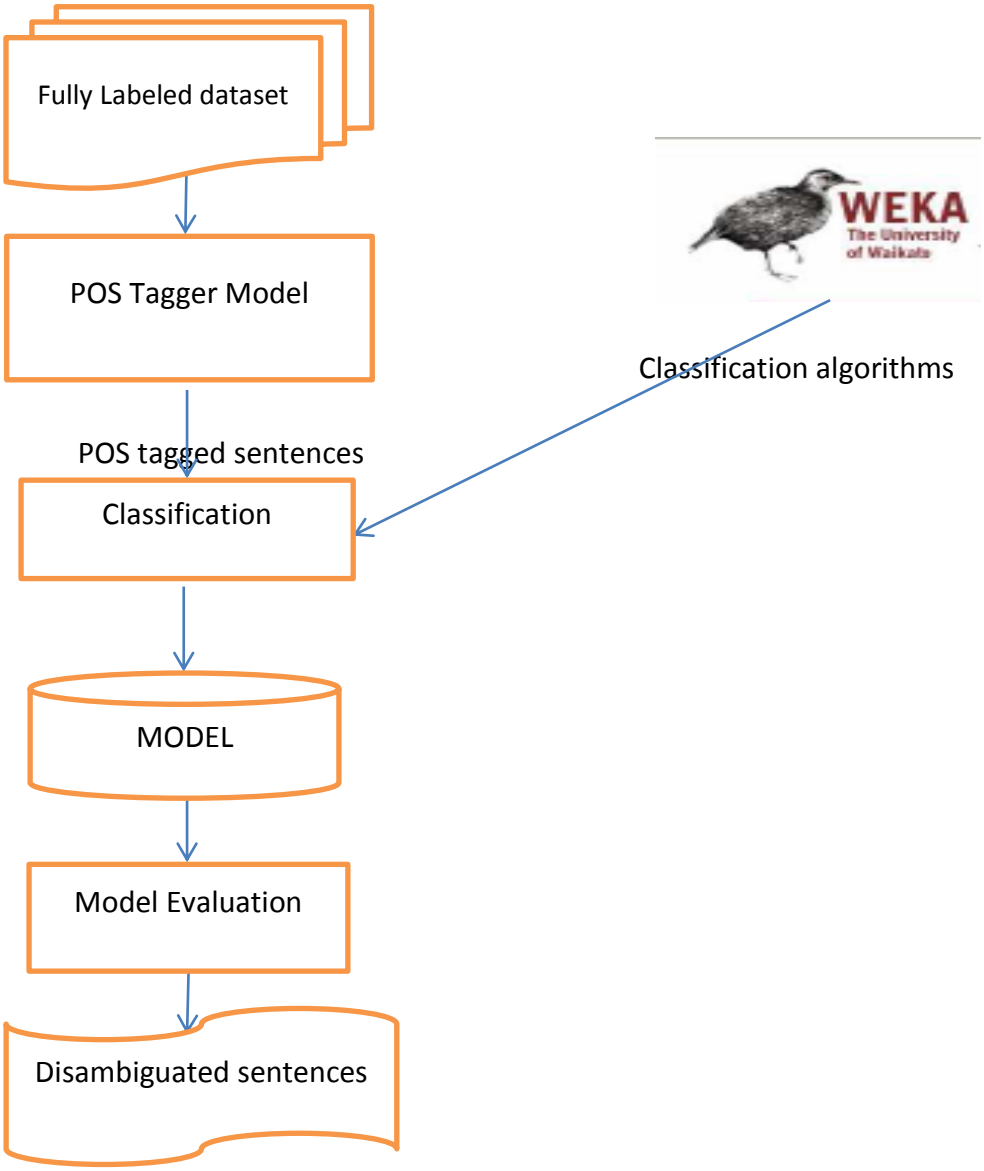


Figure-4.1 Proposed System Architecture when POS tag information is attached to fully labeled dataset

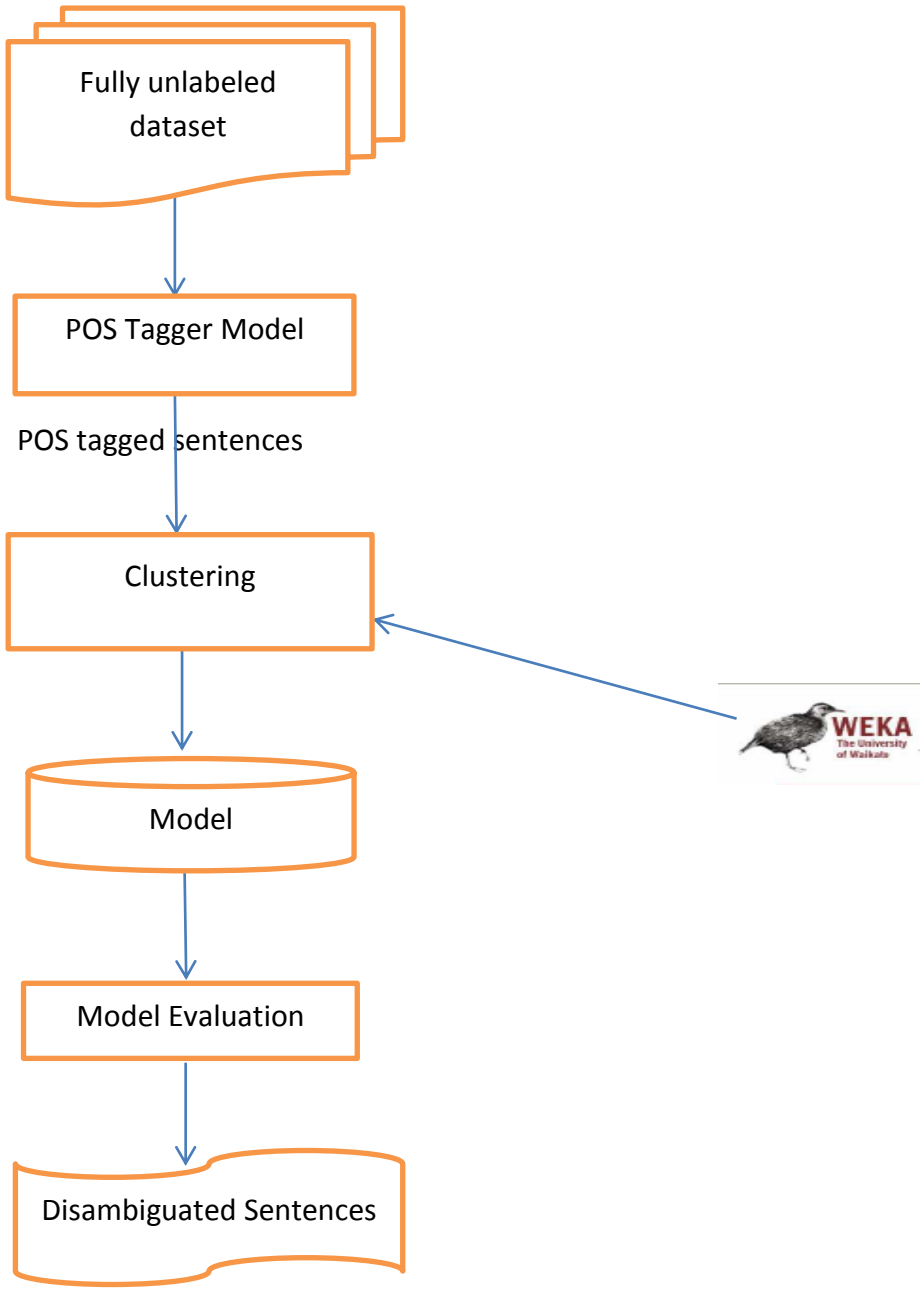


Figure-4.2 Proposed Architecture of the system for unsupervised learning method using POS tagged corpus

As it is seen in figure-4.2, the architecture above accepts fully unlabeled corpus and given for the POS tagger then the POS tagged corpus is lastly given to clustering algorithms to get fully labeled corpus.

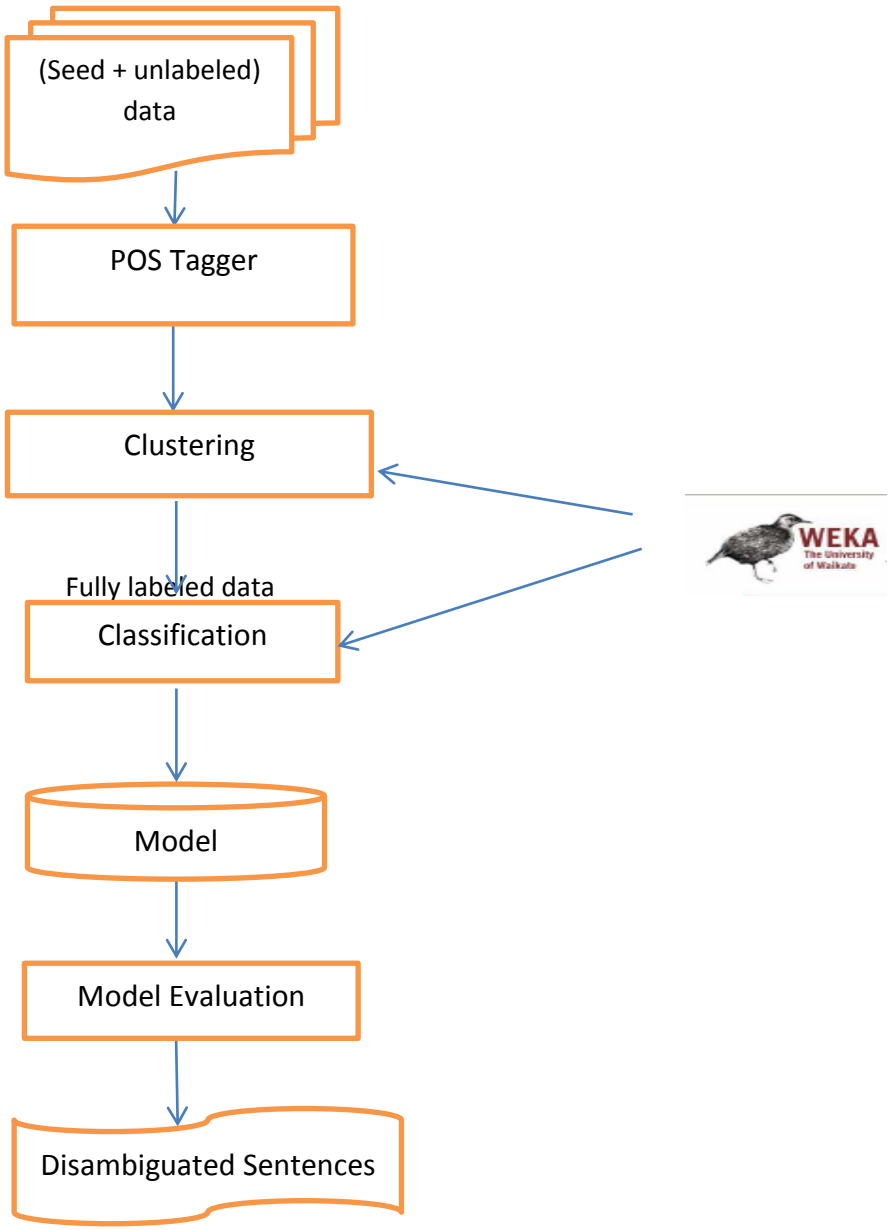


Figure-4.3 Proposed Architecture of the system when POS tagged information is involved on few labeled (seed) and unlabeled data

As it is seen in figure-4.3 above, the system accepts few seed data together with many unlabeled data. It is seen firstly given to the POS tagger after that it is given to clustering algorithms to be fully labeled. Lastly, the fully labled corpus is given to the classification algorithms to get disambiguated sentences.

4.4 Techniques

As it has been mentioned above, the data which was acquired from the previous researcher involve annotated seed examples together with the fully labeled data resulted after clustering. Even if there are three ways of selecting Seed examples, the already chosen seed examples used by [32] using labeling of single defining collocates for each class and use of salient corpus collocates has been adopted with no modification. In addition, the first task done in this study is obtaining a benchmark for WSD prototype development using five algorithms (AdaboostM1, Bagging, ADtree, SMO and Naïve Baye).

In this study, the researcher was focused on investigating the application of POS tagged corpus for experimenting three machine learning paradigms using clustering and classification algorithms for building and evaluating the WSD model.

4.5 Tools

4.5.1 WEKA

As it has been mentioned above, machine learning algorithms AdaboostM1, Bagging, ADtree, SMO and Naïve Bayes are selected to be used in this experiment and hence the tool (Weka-3.6.11) which encompasses all the above algorithms is considered worthy to be used in this study as well. The tool has been chosen to be used in this experiment, due to its free availability and beyond the researcher's friendliness to use it.

4.5.2 Conditional Random Fields (CRF++) Tool

CRF++ is a simple, customizable and open source implementation of Conditional Random Fields for segmenting/labeling sequential data. It can be applied on a variety of NLP tasks such as text chunking, NER, IE, POS and concept chunking [85]. It is developed in C++ and used less memory during training and testing. This tool has been chosen to be used in this experiment due to the available POS tagger model for Amharic already developed by [84]. Hence, the researcher has got the tool easily useable for POS tagging of the corpus to be used in the experiment.

4.6 Evaluation Technique

In this study two evaluation metrics have been used since clustering as well as classification algorithms have been applied. Each of the evaluation measures have been considered in the next subsection.

4.6.1 Classification performance Measures

In this study as classification algorithms are applied on the dataset which has been fully labeled with the use of clustering algorithms used at the start. As all the algorithms used in this study are available in weka package, there are some modes of evaluation available to be set in the package. Among the available modes like use training set, percentage split and 10-fold cross validation; 10-fold cross validation has been used in this study. As most experiments employ random choice strategy, 10-fold cross validation has also been used for this study in that it iteratively experiments after dividing the dataset into ten mutually exclusive folds then it averages the results obtained in each trial. The number of folds has been used as ten in that it is a default in weka. Hence, during the experiment, the dataset is initially divided into ten parts (folds) then averaging the results will be done after holding out each part in turn. In 10-fold cross validation, each data point will be used once for testing and 9 times for training.

In order to measure the performance of a binary classification tasks confusion matrix (or error matrix has been applied. Confusion matrix consists of columns and rows that list the number of instances as absolute or relative ("actual class" versus "predicted class") ratios [86]. The following figure shows confusion matrix by taking P as a label of class one and N as label of second class.

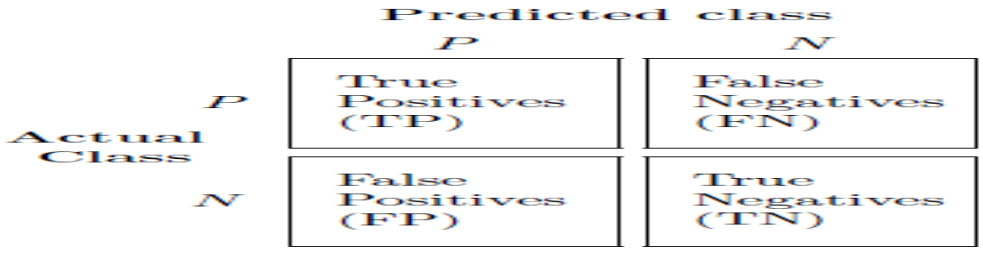


Figure-4.4 A two class confusion matrix adopted from [86]

As seen in figure-4.3 above, TP and FP rates are performance metrics that are especially useful for imbalanced class problems. TP rate provides useful information about the fraction of positive (or relevant) samples that were correctly identified out of the total pool of positives unlike that of FP rates which are negative instances that are incorrectly labeled. On the other hand, TN provides information about true instances negatively labeled and FN provides information about false instances that were incorrectly labeled.

The other performance metrics are precision, recall, F-score, prediction error and accuracy. The brief description of each performance metrics is presented as below:

Precision and Recall are metrics that are commonly used in information technology. Both measures are based on False and True positive rates.

Recall is synonymous to true positive rate and also sometimes called sensitivity. In WSD, it refers to the percentage of correctly identified senses out of the total senses [86][64]

Precision refers to the percentage of correctly identified senses out of the total [64]

F-measure is computed as a combination of recall and precision.

Prediction Error: is computed as a ratio of false predictions out of the total predictions.

Accuracy: is a performance metric used for WSD systems and is computed as a ratio of sum of correct predictions to the total number of prediction done by the system [86][32].This performance has also been used for the measure of binary classifiers. The computation is done as follows:

$$ERR = \frac{FP + FN}{FP + FN + TP + TN} = 1 - ACC$$

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} = 1 - ERR \quad (4.1)$$

4.6.2 Clustering performance Measures

Like evaluation measures of classification algorithms, two modes of evaluation (percentage and class to cluster) are available for clustering algorithms. Among the two modes, class to cluster has been used in this study. According to [87], while class to cluster mode of evaluation is used weka assigns classes to the clusters based on the majority value to the class attribute within each cluster in test phase. Lastly, weka presents a corresponding confusion matrix after computing clustering error. The above measure hence provides the measure of how well it has been able to generalize the clustering result.

CHAPTER FIVE

5. EXPERIMENTATION AND DISCUSSION

As discussed in the previous chapters, the theme of this chapter is to experiment the performance improvement that could be brought with the use of POS tagged data for improving the accuracy of WSD prototype model. In this section, three of the machine learning paradigms (Supervised, Unsupervised and semi-supervised) were experimented using two clustering algorithms (EM and Simple K-means) and five classification algorithms (AdaboostM1, Bagging, ADtree, SMO and Naïve Bayes). In addition, the experimental procedure together with the analysis of the results is presented in this section.

5.1 Experimentation Procedure

In this study, four experimental procedures were followed in general. These are:

- To obtain Benchmark result for WSD prototype development using Amharic dataset
- To investigate the application of POS tagged corpus to WSD prototype development using supervised machine learning method
- To investigate the application of POS tagged corpus to WSD prototype development using unsupervised machine learning method
- To investigate the application of POS tagged corpus to WSD prototype development using semi-supervised machine learning method

I. Experiment to obtain Benchmark Result

Firstly, the researcher has experimented to get benchmark results formerly achieved by previous researcher [32]. Moreover, the researcher believes it is worth to obtain this benchmark since it will be good enough to do the comparison and to see the effect of POS tag information at last. The experiment to attain benchmark result was carried out on:

i. Determining the effect of seed example on WSD prototype development

ii. Comparison of supervised, unsupervised and semi-supervised algorithms on Amharic dataset

iii. Identification of best performing algorithm suitable for WSD prototype development

iv. Investigation of the optimal context window size enough for determining disambiguation on Amharic dataset

A. Effect of training seed options on Amharic WSD prototype

Bench Mark using seed examples		
Algorithm	1 seed word	2 and 3 seed words For each senses avg.
Adabostm1	84.52%	81.49%
Bagging	79.80%	76.69%
ADtree	87.70%	88.66%
SMO	87.40%	84.32%
Naïve Bayes	38.22%	41.55%

Table-5.1 Benchmark performance variants using different size seed words

As it could be seen from Table-5.1 above, four of the algorithms (AdaboostM1, Bagging, ADtree and SMO) have resulted better performance score for one seed word for each sense. The performance of AdaboostM1, Bagging, ADtree and SMO algorithms were 84.52%**,** 79.80%**,** 87.70% and 87.40% respectively.

In addition, Naïve Bayes has resulted poor performance scores in both cases due to the probabilistic theory that intuitively uses independent assumption besides the nature of the algorithm to perform well for large dataset with balanced distribution. The performance score achieved by Getahun [32] was 41.55% for Naïve Bayes respectively. Hence, the researcher believes the experimental setup currently used for determining effect of seed word has been confirmed to be as similar as that of Getahun [32].

B. Comparison of supervised, semi-supervised and unsupervised learning methods

i. Experimental Benchmark result obtained from supervised learning

supervised Machine Learning Algorithms					
Training Wordset	AdaboostM1	Bagging	Adtree	SMO	Naïve Bayes
Atena	58.60%	71.63%	59.54%	83.72%	84.65%
Derese	70.53%	63.77%	81.64%	81.16%	84.06%
Tenesa	52%	60%	52%	67.5%	64%
Ale	52.40%	56.73%	53.37%	68.75%	75%
Bela	59.80%	67.84%	61.81%	76.38%	78.89%
Average	58.67%	63.99%	61.67%	75.50%	77.32%

Table-5.2 Benchmark result using supervised learning algorithms

The benchmark performance score of machine learning algorithms using fully labeled dataset (adopted from [32] has resulted similar result like that of Getahun's former experimental result. As it could be seen in Table-5.2 above, Bagging algorithm have resulted an average score of 63.99% unlike that of Getahun's [32] score of 63.99%. The other average scores obtained by this benchmark experiment for supervised learning are exactly the same as that of Getahun [32].

In addition, Naïve Baye algorithm is the one which have resulted better average performance score (77.32%) on fully labeled data set even though SMO has also resulted a comparable score of 75.50%.

ii. Experimental Benchmark result obtained from unsupervised learning

Bench Mark using unsupervised Machine Learning Algorithms		
Training Wordset	EM	K-MEANS
Atena	61.4%	57.21%
Derese	62.80%	64.25%
Tenesa	54.5%	53%
Ale	60.1%	52.40%
Bela	56.29%	50.75%
Average	59.02%	55.52%

Table-5.3 Benchmark result obtained using Unsupervised Learning

As it could be seen from the benchmark result in Table-5.3 above, performance score of more than 60% has been achieved using clustering for the words 'atena', 'derese' and 'ale' using EM

algorithm like that of Getahun's experimental result. Among the two clustering algorithms, EM algorithm has scored average performance of 59.02% and simple K-means has scored 55.52%. The average experimental performance score achieved by Getahun was 60.11% and that of the baseline score is 59.02% using EM. Still other experiments deemed to be done so as to improve the performance score of clustering.

iii. Experimental Benchmark result obtained from semi-supervised learning

Bench Mark using SEMI-supervised Machine Learning Algorithms					
Training Wordset	AdboostM1	Bagging	ADtree	SMO	Naïve Bayes
Atena	93.95%	87.44 %	97.21 %	90.70%	39.53%
Derese	90.34%	76.81 %	95.65%	92.75%	76.14 %
Tenesa	82%	82%	83.50%	82%	32%
Ale	63.94%	64.42%	74.52%	84.13%	65.38%
Bela	89.45%	85.43%	91.46%	87.44%	27.14%
Average	83.94%	79.22%	88.47%	87.40%	48.04%

Table-5.4 Benchmark result using semi-supervised Learning

The benchmark result obtained in Table-5.4 above has shown that the performance scores for ADtree, SMO and AdaboostM1 are 88.47%, 87.40% and 83.94% respectively; which actually was exactly the same as that of Getahun's score. For the other two algorithms (Bagging and Naïve Bayes) the performance score obtained was 79.22% for Bagging and 48.04% for Naïve Bayes which has shown even a better result though the performance achieved was not greater than a percent difference.

C. Benchmark result obtained during comparison of algorithms

Algorithm	Benchmark performance	Running time in seconds
AdaboostM1	83.94%	0.036
Bagging	79.2207%	0.044
ADtree	88.47%	0.012
SMO	87.40%	0.356
Naïve Bayes	39.09%	-

Table-5.5 Benchmark result of algorithms with their performance score and running time

D. Benchmark result obtained to determine optimal context window size

Window Size using AdaboostM1					
	Atena	Derese	Tenesa	Ale	Bela
one-one	93.49 %	90.34 %	82 %	64.42 %	89.45 %
two-two	93.95 %	90.34 %	82 %	68.27 %	89.45 %
three-three	93.95 %	90.34 %	82 %	68.27 %	89.95 %
four-four	93.95 %	90.34 %	84 %	68.27 %	89.95 %
five-five	93.95 %	90.34 %	84 %	66.83 %	89.95 %
six-six	93.95 %	90.34 %	84 %	63.94 %	89.95 %
seven-seven	93.95 %	90.34 %	84 %	63.46 %	89.45 %
eight-eight	93.95 %	90.34 %	84 %	64.42 %	89.45 %
nine-nine	93.95 %	90.34 %	83%	63.94 %	89.45 %
ten-ten	93.95 %	90.34 %	82 %	63.94 %	89.45 %

Table-5.6 Benchmark result obtained for window size determination using AdaboostM1

As it could be seen from Table-5.6 above, the benchmark result obtained for determining the optimal context window size using AdaboostM1 algorithm have repeated exactly the same result as that of the former researcher. The result has shown that window size of 2-2 is enough for the experiment involving the ambiguous words ‘atena’ and ‘ale’. Window size resulted for ‘derese’, ‘tenesa’ and ‘bela’ is 1-1, 4-4 and 3-3 respectively and lastly the average is window size using this algorithm has been found out to be 3-3.

Window Size using Bagging					
	Atena	Derese	Tenesa	Ale	Bela
one-one	96.74 %	94.69 %	82.5 %	67.31 %	91.46 %
two-two	89.77 %	86.47 %	81.5 %	70.19 %	89.45 %
three-three	88.84 %	81.64 %	81.5%	67.31 %	86.93 %
four-four	87.44 %	81.64 %	84%	65.38 %	86.43 %
five-five	87.44 %	79.23 %	81.5%	67.79 %	86.43 %
six-six	87.44 %	76.33 %	81%	67.79 %	86.43 %
seven-seven	87.44 %	76.33 %	81.5%	67.79 %	84.92 %
eight-eight	86.98 %	78.26 %	81%	64.90 %	85.43 %
nine-nine	86.51 %	78.26 %	82%	64.42 %	85.43 %
ten-ten	87.44 %	78.26 %	82%	64.42 %	85.43 %

Table-5.7 Benchmark Result obtained for window size determination using Bagging Algorithm

The benchmark result in Table-5.7 above has shown that for Bagging algorithm a window size of 1-1 to be enough for the words: ‘atena’, ‘derese’ and ‘bela’. The algorithm has resulted window size of 4-4 and 2-2 to be enough for the words ‘tenesa’ and ‘ale’. The benchmark result has shown that window size of 1-1 to be enough for the word ‘bela’ unlike Getahun’s experimental results of 2-2 window for this word only.

Window Size using ADtree					
	Atena	Derese	Tenesa	Ale	Bela
one-one	97.21 %	94.69 %	81.5%	63.46 %	91.96 %
two-two	97.21 %	95.65 %	81.5%	73.08 %	91.96 %
three-three	97.21 %	95.65 %	81.5%	76.92 %	90.95 %
four-four	96.74 %	95.65 %	83.5%	77.88 %	90.95 %
five-five	96.74 %	95.65 %	83.5%	80.29 %	93.47 %
six-six	96.74 %	95.65 %	83.5%	78.37 %	93.47 %
seven-seven	96.74 %	95.65 %	83.5%	76.92 %	92.46 %
eight-eight	96.74 %	95.65 %	83.5%	76.44 %	90.45 %
nine-nine	96.28 %	95.65 %	83.5%	76.44 %	90.45 %
ten-ten	96.28 %	95.65 %	83.5%	74.52 %	91.46 %

Table-5.8 Benchmark result obtained using ADtree algorithm for window size determination

The benchmark result obtained in Table-5.8 above has shown that, a window size of 1-1, 2-2, 4-4 is enough for the words 'atena', 'derese' and 'tenesa' respectively using ADtree algorithm. The window size resulted to be enough for this algorithm is 5-5 using 'ale' and 'bela' words. The overall benchmark result is exactly the same as that of Getahun's [32] experimental result.

Window Size using SMO					
	Atena	Derese	Tenesa	Ale	Bela
one-one	96.74 %	95.17 %	83.5%	70.193 %	93.47 %
two-two	96.28 %	94.69 %	82%	75%	91.46 %
three-three	95.35 %	93.72 %	82%	79.33 %	90.95 %
four-four	95.35 %	93.72 %	82%	80.77 %	90.95 %
five-five	94.88 %	93.24 %	82%	82.21 %	90.45 %
six-six	94.42 %	93.24 %	82%	84.13 %	88.95 %
seven-seven	94.42 %	92.75 %	82%	84.62 %	87.94 %
eight-eight	93.49 %	92.75 %	82%	86.06 %	87.44 %
nine-nine	92.09 %	92.75 %	82%	84.62 %	87.44 %
ten-ten	90.70 %	92.75 %	82%	84.13 %	87.44 %

Table-5.9 Benchmark result obtained for window size determination using SMO algorithm

The benchmark result obtained using SMO algorithm has resulted with a 1-1 window size for all of the ambiguous words except 'ale' which has resulted window size of 8-8 to be required as seen in Table-5.9 above. Totally the result obtained in this benchmark experiment is the same as that of Getahun's [32] experimental result.

Window Size using Naïve Bayes					
	Atena	Derese	Tenesa	Ale	Bela
one-one	95.35 %	93.72 %	83.5%	69.71 %	92.46 %
two-two	71.63 %	60.87 %	63.5%	69.71 %	69.35 %
three-three	59.07 %	48.79 %	48.5%	71.63 %	54.27 %
four-four	46.05 %	43.48 %	38%	73.08 %	43.22 %
five-five	42.33 %	39.13 %	36%	71.63 %	38.69 %
six-six	41.4 %	35.75 %	35%	70.67 %	35.68 %
seven-seven	40%	35.27 %	34.5%	68.75 %	35.68 %
eight-eight	39.53 %	34.78 %	34.5%	68.75 %	32.16 %
nine-nine	39.53 %	33.33 %	34.5%	65.87 %	27.64%
ten-ten	39.53 %	31.40 %	32%	65.38 %	27.14 %

Table-5.10 Benchmark result on window size using Naïve Bayes algorithm

The benchmark result presented above in Table-5.10 has shown an exact same result as that of Getahun's [32] . The window size of 1-1 was found out to be enough for all of the ambiguous words except 'ale' which has required a window size of 4-4. The average optimal window size obtained using Naïve Bayes algorithm was 2-2 which is the same as the one reported by Getahun [32].

II. **Experiment using supervised machine learning paradigm**

In this experiment a fully labeled corpus acquired from the previous researcher has been used after only attaching POS tag information on it. Supervised learning requires annotated training data from which a classifier is induced by the algorithm as of [88]. These algorithms rely on large amounts of accurately sense annotated data to yield good results though requires high cost and time to be incurred. Supervised methods of WSD have been considered to yield poor performance while lack of enough sense tagged training data have been faced during training

phase and is called knowledge acquisition bottleneck. During this study the data which was fully labeled by the previous researcher have been used after attached POS tag information to each word.

Supervised Learning					
Wordsets	AdaboostM1	Bagging	ADtree	SMO	Naïve Bayes
Atena	75.35 %	80%	78.60 %	80%	85.12%
Derese	73.91 %	65.70%	77.29 %	74.88%	80.68%
Tenesa	50%	61%	51%	48.50%	58%
Ale	56.73%	60.58 %	52.88 %	60.58%	71.63%
Bela	71.36 %	76.38 %	69.85 %	72.36%	79.90%
Average	65.47%	68.73%	65.93%	67.26%	75.07%

Table-5.11 Result of supervised learning using a corpus with POS tag information

Discussion:

As it could be seen from experimental result on Table-5.11, Naïve Bayes algorithm has resulted better performance score of 75.07% unlike the degraded performance score of the other bootstrapping algorithms (ADtree, AdaboostM1 and Bagging) and SVM algorithm (SMO). The average performance score of AdaboostM1, Bagging, ADtree and SMO was 65.47%, 68.73%, 65.93% and 67.26% respectively. The better average performance score of Naïve Baye's algorithm is due to its capability to consider all instances of the dataset unlike starting from the seed examples of bootstrapping algorithms. In general, the performance score of supervised learning using a dataset with POS tag information has resulted an average accuracy of 75.07% but it is less as compared to the performance score achieved during the benchmark experiment which is 77.32%. The decrease in performance might have resulted due to the small size corpus used for the experiment and due to the nature of supervised methods to be limited to small contexts [9].

III. Experiment using unsupervised machine learning paradigm

Unsupervised machine learning method for developing WSD systems benefits a lot as it alleviates the knowledge acquisition bottleneck faced by supervised methods. In spite of the benefit over supervised methods, these methods have a problem as they work based on the

idea that the same sense of a word will have similar neighboring words. These methods are able to induce word senses from input text by clustering word occurrences rather than relying on shared inventory of senses and hence yield less performance unlike that of the supervised classes of machine learning methods.

Unsupervised Learning Method		
Training Data sets	EM	Simple K-means
Atena	64.19%	52.56%
Derese	57.49%	69.57%
Tenesa	54.5%	54.5%
Ale	54.81%	51.44%
Bela	55.28%	52.26%
Average	57.25%	56.07%

Table-5.12 Experimental Result of unsupervised algorithms using POS tagged corpus

As it could be seen from Table-5.12 above, the experimental result obtained using clustering algorithms (EM and Simple K-means) has been presented. The experimental result obtained has shown that the average performance score of EM was 57.25% and that of Simple K-means was 56.07%. The average performance scores which were obtained were less as compared to the benchmark with about 1.76% for EM but it has shown performance improvement for simple K-means with about 0.54%. The overall performance of the clustering has shown no significant improvement while POS tag information is involved.

IV. Experiment using Semi-supervised machine learning paradigm

In this section, the researcher has used the seed examples chosen by [32] and the POS tag information for clustering by use of two clustering algorithms (i.e EM and Simple K-means) then after fully labeled data already obtained using clustering algorithms was used for the experiment using the five machine learning algorithms: AdaboostM1, Bagging, SMO, ADtree and Naïve Bayes. In this part, four experiments have been carried out.

The first experiment was conducted to check the effect of training seed word options with POS tag information on Amharic WSD prototype development.

The second experiment was conducted to compare the supervised, semi-supervised and unsupervised machine learning algorithms using the dataset with POS tag information.

The third experiment was conducted to see the performance of the selected machine learning algorithms when the dataset was attached with POS tag information since it results with an impact on the increase with number of instances.

Lastly, the fourth experiment was conducted to investigate the effect of different context sizes on disambiguation accuracy for Amharic ambiguous word and to find out window size enough for Amharic while smaller data size has been used during the experiment and additionally when POS tag information is attached in particular.

A. Effect of training seed word Options on Amharic WSD prototype model Development

There are three ways for the selection of seed examples, which actually are dependent on the seed words which represent sense of instances. The three strategies which are available are: Use of dictionary for choosing sense of the ambiguous word, the second alternative is use of single defining collocated word for each class and the third is use of salient corpus collocates [44]. Among the three strategies Getahun [32] has used only the second and the third strategies for the selection of seed examples due to the absence of dictionary meaning for using the first strategy.

As it has already been described above, only POS tag information has been attached to the already available corpus by [32] and hence seed options which were selected tagged for this experiment. The experiment has been carried out using one seed word and average of two and three seed word options.

Seed Word Options on WSD prototype development				
Algorithm	1 seed word for each sense	2 seed word for each sense	3 seed word for each sense	Average of 2 and 3 seed words
AdaboostM1	90.96%	89.04%	89.57%	89.31%
Bagging	80.79%	79.58%	78.23%	78.90%
ADtree	92.03%	92.04%	89.11%	90.57%
SMO	89.92%	88.33%	88.6%	88.46%
Naïve Bayes	60.62%	57.14%	60.00%	58.57%

Table 5.13 Performance of seed word option using classification machine learning algorithms

Discussion:

As it could be inferred from Table-5.13, experimental result on effect of training seed word options for building model to WSD, two of the algorithms (ADtree and AdaboostM1) have resulted encouraging results with accuracy of 92.03% and 90.96% respectively for one seed word option. Bootstrapping and SVM algorithms have the capability to tackle the problem of over fitting by adding more unlabeled data. Of the above three bootstrapping algorithms (AdaboostM1 and ADtree), AdaboostM1 have scored encouraging result.

B. Experimental Result using Semi-supervised Learning Methods

Semi-supervised Learning methods					
Training data set	AdaboostM1	Bagging	ADtree	SMO	Naïve Bayes
Atena	91.16 %	85.58 %	94.4186 %	93.4884 %	51.16 %
Derese	93.72 %	76.81 %	94.69 %	90.82 %	50.24 %
Tenesa	100 %	96.5 %	99 %	97.5 %	72 %
Ale	91.35 %	82.69 %	92.31 %	89.42 %	54.33 %
Bela	85.43 %	63.32 %	82.91 %	78.39 %	75.38 %
Average	92.33%	80.98 %	92.67 %	89.93%	60.62%

Table 5.14 Semi-supervised result using fully labeled data using EM algorithm

Semi-supervised Learning methods					
Training data set	AdaboostM1	Bagging	ADtree	SMO	Naïve Bayes
Atena	83.72 %	86.98 %	86.51 %	86.05 %	49.30 %
Derese	92.27 %	71.01 %	91.79 %	86.47 %	42.51 %
Tenesa	80%	80.5 %	86.5 %	82.5 %	33.5 %
Ale	90.38 %	89.42 %	92.31 %	91.83 %	61.06 %
Bela	92.97 %	89.45 %%	91.96 %	89.45 %	34.17 %
Average	87.87%	83.47%	89.81%	87.26%	44.11%

Table-5.15 Result of semi-supervised learning method using clustering by k-means

As it could be seen from Tables 5.14 and 5.15 above, the experiment has been carried out using two alternative corpuses tagged with POS information. Clustering of POS tag information to the corpus has been done using EM and Simple K-means. Hence, two experimental results have been obtained as seen above.

The first experimental result in Table-5.14 above is obtained using a corpus which has been tagged with POS information but the clustering is done using EM algorithm. The result obtained in this experiment is encouraging as the performance scores obtained are: 92.67% for ADtree, 92.33% for AdaboostM1, 89.93% for SMO, 80.98% for Bagging and 60.62% for Naïve Bayes algorithm.

The second experimental result presented in Table-5.15 above have been obtained using a corpus which has been POS tagged using Simple K-means algorithm. The performance score obtained using this corpus is : 89.81 % using ADtree, 87.87% using AdaboostM1, 87.26% using SMO, 83.47% using Bagging and 44.11% using Naïve Bayes algorithm

The overall result obtained using semi-supervised learning using the corpus which has been tagged with POS information using EM algorithm has resulted better performance score of 92.67% using ADtree and 92.33% using AdaboostM1 unlike the bench mark result of 88.47% using ADtree and 83.94% using AdaboostM1 algorithms.

C. Experiment on window size determination for WSD prototype development using POS tagged corpus

Languages which are well resourced have a standard window size, number of surrounding contexts enough for WSD have been determined through researches of which English is one. According to [89], the standard window size which was deemed enough for English language was two-two on either side of the ambiguous word. Despite the standard widow size for English, till recently there is no standard window size determined to be enough for WSD system but some research works for Amharic WSD have experimented was limited to only a maximum of ten-ten from either side of the target word and have not been reached to some standard till recently.

Some of the research works done to determine the window sizes enough for Amharic language are:

Solomon [30] has obtained a three-three window size to be enough for WSD study using supervised machine learning methods. The accuracy which has been achieved was (70-83) % on five of the ambiguous Amharic words and data size of 1045 sentences.

Solomon [31] was the second to experiment the window size enough for WSD using Unsupervised machine learning algorithms and obtained two-two to three-three window size to be enough depending on the algorithm used. During the experiment on the same five ambiguous Amharic words window size of two-two on either side of the target word was considered enough for WSD using agglomerative single link and complete link which resulted accuracy in the range of (51.9-71.1) %. Window size of three-three was considered effective during his experiment using EM and Simple K-means clustering algorithms where accuracies in the range of (65.1-75.9) % were achieved.

The next researcher who has experimented on window size determination research work on Amharic WSD was [32]. Getahun [32] has experimented the window size in the range of one-one to ten-ten on either size of the target word using semi-supervised machine learning algorithms involving 1031 sentences. Similarily, Getahun [32] has obtained a two-two window

size to be enough for Naïve Bayes algorithm and a three-three window size to be enough using bootstrapping and SMO algorithms to build WSD systems.

Lastly, [34] has experimented on window-size enough for WSD using Ensemble of Naïve Bayes. Birhane [34] has used a total of about 1415 sentences and reached to a conclusion that a window size of three-three is enough for the building WSD system using Ensemble of Naïve Bayes.

Window Size using AdaboostM1					
	Atena	Derese	Tenesa	Ale	Bela
one-one	90.23 %	78.26 %	100%	81.25 %	62.81 %
two-two	91.16%	78.74 %	100%	81.73 %	63.32 %
three-three	89.77 %	78.74 %	100%	80.77 %	65.83 %
four-four	90.7 %	95.17 %	100%	80.77 %	66.33 %
five-five	89.30 %	94.69 %	100%	88.46 %	69.85 %
six-six	91.16 %	94.69 %	100%	89.42 %	68.34 %
seven-seven	90.23 %	93.72 %	100%	89.42 %	68.34 %
eight-eight	91.63 %	94.69 %	100%	87.98 %	63.82 %
nine-nine	91.16 %	93.72 %	100%	91.83 %	64.32 %
ten-ten	90.7 %	94.20 %	100%	92.79 %	85.43 %

Table-5.16 Experimental result of AdaboostM1 algorithm for window size determination

As it could be seen from the above experimental result in Table-5.16, for AdaboostM1 algorithm window size of 8-8 for ‘atena’ and 10-10 for the words ‘ale’ and ‘bela’ has been seen to score better. For ‘derese’ a 4-4 window size has been considered to be enough but a window size of 1-1 is seen for ‘tenesa’ even though the accuracy achieved is 100% in all cases which shows existence of common context in all window size for ‘tenesa’. The accuracy achieved for ‘atena’ is 91.63% but that of ‘ale’ and ‘bela’ is 92.79% and 85.43% respectively.

The performance score of ‘ale’ lies in the range of 80.77 % to 92.79 % which is by far better accuracy in all the window size as compared to the previous lower performance score achieved by Getahun [32] which was 63.46 % - 68.27 % which might have been due to the effect of POS tag information applied on the dataset.

Therefore, the average window size using AdaboostM1 algorithm is 6.6 shows window size of seven is needed while POS tag information is added for experiment of WSD prototype development. In addition, except 'tenesa' and 'ale' a decrease in performance scores have been seen on the data using AdaboostM1 algorithm which may even be due to the POS tag information added on each attribute since POS tag my result with an increase in the number of attributes, a cause for over fitting.

Window Size using Bagging					
	Atena	Derese	Tenesa	Ale	Bela
one-one	93.95 %	72.95 %	100%	79.33 %	64.82 %
two-two	93.95 %	74.4 %	96.5%	82.7 %	63.32 %
three-three	87.91 %	73.91 %	96.5%	81.73 %	60.80 %
four-four	83.72 %	75.36 %	96.5%	80.77 %	60.80 %
five-five	83.72 %	75.85 %	96.5%	82.21 %	63.32 %
six-six	83.72 %	77.29 %	96.5%	82.69 %	60.30 %
seven-seven	83.72 %	77.78 %	96.5%	83.17 %	60.30 %
eight-eight	84.65 %	76.33 %	96.5%	83.17 %	59.8 %
nine-nine	84.19 %	75.36 %	96.5%	82.21 %	61.81 %
ten-ten	85.58 %	74.88 %	96.5%	82.69 %	62.31 %

Table-5.17 Experimental result of Bagging algorithm for window size determination

As it could be seen from Table-5.17 above, the window size of 1-1 has been seen for 'atena', 'tenesa' and 'bela' using Bagging algorithm with performance score of 93.95, 100% and 64.82 respectively. In addition, window size of 7-7 and 2-2 has been seen for the words 'derese' and 'ale' with accuracies of 77.78% and 82.69% respectively. The average window size hence for Bagging algorithm from the above experimental result is 2.4 which is 3. Hence, a window size of 3-3 is required for Bagging algorithm averagely.

The average window size for Bagging algorithm with the incorporation of POS tag information has not result other window size as that of the bechmark. But eventhough, the window size is not different from the benchmark the accuracy has shown a great decrease in all of the words.

Window Size using ADtree					
	Atena	Derese	Tenesa	Ale	Bela
one-one	89.77 %	77.78 %	100%	81.25 %	63.32 %
two-two	92.09 %	79.71 %	100%	81.73 %	63.82 %
three-three	92.09 %	74.4 %	100%	79.81 %	64.82 %
four-four	90.7 %	93.24 %	100%	78.37 %	62.81 %
five-five	90.7 %	92.75 %	100%	88.94 %	64.82 %
six-six	88.84 %	92.75 %	99.5%	90.38 %	59.3 %
seven-seven	90.7 %	92.75 %	99%	89.90 %	65.33 %
eight-eight	91.63 %	92.27 %	99%	89.90 %	55.78 %
nine-nine	91.63 %	92.27 %	99%	91.35 %	56.28 %
ten-ten	90.23 %	91.79 %	99%	91.83 %	83.92 %

Table-5.18 Experimental result of ADtree algorithm for window size determination

As seen from Table-5.18 above, window size of 1-1, 2-2 and 4-4 has been seen for the words ‘tenesa’, ‘atena’ and ‘derese’ respectively using ADtree algorithm. In addition, window size of 10-10 has been seen for the words ‘ale’ and ‘bela’ with accuracies of 91.8269% and 83.9196% respectively. The average window size using this algorithm is hence 5.4 which is 6 words from either side of the target word. The average optimal widow size of 6-6 shows an increase unlike that of 4-4 window size achieved in the bench mark experiment. The performance scores also result with less performance except for ‘tenesa’ and ‘ale’.

Window Size using SMO					
	Atena	Derese	Tenesa	Ale	Bela
one-one	93.95 %	74.88 %	100%	77.88 %	66.83 %
two-two	93.02 %	76.81 %	100%	80.77 %	66.83 %
three-three	94.88 %	76.33 %	97.5%	81.25 %	63.32 %
four-four	94.42 %	94.2 %	97.5%	82.69 %	65.83 %
five-five	94.88 %	92.75 %	97.5%	87.02 %	64.32 %
six-six	93.95%	92.27 %	97.5%	86.54 %	64.82 %
seven-seven	93.95 %	90.82 %	97.5%	87.02 %	63.82 %
eight-eight	93.95 %	90.34 %	97.5%	87.98 %	59.8 %
nine-nine	94.42 %	89.86 %	97.5%	88.94 %	61.81 %
ten-ten	93.49 %	90.82 %	97.5%	90.38 %	77.89 %

Table-5.19 Experimental result of SMO algorithm for window size determination

The experimental result presented in Table-5.19 above shows that window size of 1-1, 3-3 and 4-4 has been achieved for ‘tenesa’, ‘atena’ and ‘derese’ with accuracies 100%, 94.88% and 94.20% respectively using SMO algorithm. In addition, ‘ale’ and ‘bela’ has resulted with window size of 10-10 with accuracies of 90.39% and 77.89% respectively. The average window size resulted is 5.6 which is 6 words on either side of the target word.

Window Size using Naïve Bayes					
	Atena	Derese	Tenesa	Ale	Bela
one-one	89.30 %	60.87 %	94%	70.19%	62.81 %
two-two	84.65 %	56.04 %	88%	65.39 %	67.34 %
three-three	78.61 %	81.16 %	90%	59.14 %	62.81 %
four-four	70.7 %	81.16 %	84%	54.33 %	65.33 %
five-five	65.12%	75.36 %	83.5 %	64.90 %	68.34 %
six-six	57.67 %	69.57 %	81%	62.02 %	70.35 %
seven-seven	54.42 %	62.80 %	80%	58.65 %	67.84 %
eight-eight	53.02 %	58.45 %	79.5%	52.40%	68.84 %
nine-nine	51.16 %	52.17 %	75%	57.21 %	69.35 %
ten-ten	70.53 %	50.24 %	73.5%	53.85 %	74.37 %

Table-5.20 Experimental result of Naïve Bayes algorithm for window size determination

As seen on Table-5.20 above, window size of 1-1 has been achieved for ‘atena’, tenesa’ and ‘ale’ with accuracy of 89.30%, 94% and 70.19% respectively using Naïve Bayes algorithm. In addition, for the words ‘derese’ and ‘bela’ window size of 3-3 and 10-10 has been resulted in

the experiment with accuracies 81.16% and 74.37% respectively. The average window size resulted in this experiment is 3.2 which is 4. The average window size achieved in the bench mark experiment was 2-2. So, the accuracies also achieved shows less as compared to the bench mark with greater average window size of 4-4 for the dataset equipped with POS tag information.

V. Comparison of Supervised, Semi-Supervised and Unsupervised learning methods on Amharic words data set

A. Comparison of Algorithms in Performance Score

Machine Learning type	Performance score	Algorithm
Semi-supervised	92.67%	ADtree
Supervised	75.07%%	Naïve Bayes
Unsupervised	57.25%	EM

Table-5.21 comparison of Algorithms

B. Comparison of Algorithms in Running Time

Algorithm	Performance	Running time in seconds
AdaboostM1	87.87%	0.128
Bagging	83.47%	0.47
ADtree	89.81%	0.3675
SMO	87.26%	0.286
Naïve Bayes	44.11%	-

Table 5.22 Experimental Result Obtained by Comparison of the Algorithms

5.2 Comparison of the Experimental result with the Baseline

5.3.1 Comparison of effect of seed word with baseline

Algorithm	Getahun's score	Benchmark	Experimental score using POS tag information
AdaboostM1	85.31%	84.52%	90.96%
Bagging	78.66%	79.80%	80.79%
ADtree	87.89%	87.70%	92.03%
SMO	84.13%	87.41%	89.92%
Naïve Bayes	38.22%	38.22%	60.62%

Table-5.23 comparison of effect of one seed word with Baseline

Algorithm	Getahun's score	Benchmark	Experimental score using POS tag information
AdaboostM1	81.49%	81.49%	89.31%
Bagging	77.51%	76.69%	78.90%
ADtree	88.66%	88.66%	90.57%
SMO	75.48%	84.32%	88.46%
Naïve Bayes	41.55%	41.56%	58.57%

Table-5.24 Comparison of Effect of Seed Word with Baseline

5.3.2 Comparison of Experimental Result of Algorithms with Baseline

Experimental Result of Supervised Learning					
	AdaboostM1	Bagging	ADtree	SMO	Naïve Bayes
Getahun	58.67%	64.12%	61.77%	75.5%	77.32%
Benchmark	58.66%	63.99%	61.67%	75.50%	77.32%
Experimental score using POS tag	65.469%	68.73%	65.92%	67.26%	75.07%

Table-5.25 Comparison of Experimental Result of Supervised Learning with Baseline

Unsupervised Learning Method		
	Expected Maximization	Simple k-means
Getahun	60.11%	55.52%
Benchmark	59.01496%	55.52362%
Experimental result with POS tag inf.	57.2516%	56.0653%

Table-5.26 Comparison of Experimental Result of Unsupervised Learning with Baseline

Experimental Result of Semi-Supervised Learning					
	AdaboostM1	Bagging	ADtree	SMO	Naïve Bayes
Getahun	83.94%	78.28%	8.47%	87.4%	47.9%
Benchmark	83.94%	79.90%	88.47%	87.4%	48.04%
POS using K-means	87.87%	83.47%	89.81%	87.26%	44.11%
POS using EM	92.33%	80.98%	92.67%	89.93%	60.62%

Table-5.27 Comparison of Experimental Result of Semi-supervised Learning with Baseline

5.3.3 Comparison of Optimal Window size with Baseline

Words	Comparison of Optimal Window sizes of Bootstrapping Algorithms					
	AdaboostM1		Bagging		ADtree	
	Baseline(3-3)	ER(6-6)	Baseline(3-3)	ER(6-6)	Baseline(3-3)	ER(6-6)
Atena	93.95%	91.16%	88.84%	83.72%	97.21%	88.84%
Derese	90.34%	94.69%	81.64%	77.3%	95.65%	92.75%
Tenesa	82%	100%	81.5%	96.5%	81.5%	99.5%
Ale	68.27%	88.46%	67.31%	82.69%	76.92%	90.4%
Bela	89.95%	68.34%	86.94%	60.30%	90.96%	59.3%
Average	84.90%	88.53%	81.24%	80.10%	88.45%	86.15%

Table-5.28 Comparison of Bootstrapping Algorithms with Baseline

WORDS	Comparison of Experimental result with Baseline			
	SVM		Naïve Bayes	
	Baseline(3-3)	ER(6-6)	Baseline(2-2)	ER(4-4)
Atena	95.35%	93.95%	71.63%	70.7%
Derese	93.72%	92.27%	60.87%	81.16%
Tenesa	82%	97.5%	63.5%	84%
Ale	79.33%	86.54%	89.71%	54.33%
Bela	90.96%	64.82%	69.35%	65.33%
Average	88.27%	87.02%	71.01%	71.10%

Table-5.29 Comparison of Naïve Bayes and SMO Algorithms with Baseline

Summary:

In general, the study was focused on investigation of effect of POS tag information to develop WSD prototype model using the three classes of machine learning methods (supervised, unsupervised and semi-supervised). Of the three classes of machine learning approaches, semi-supervised has resulted better performance score followed by the supervised methods and last the unsupervised methods which shows semi-supervised machine learning approach is capable to exploit unlabeled data to maximize the accuracy of WSD prototype model.

The performance score resulted while semi-supervised paradigm has been applied was: 92.67% for ADtree, 92.33% using AdaboostM1, 89.93% using SMO, 80.98% using Bagging and 60.62% using Naïve Bayes algorithm. The performance score resulted was better than all the scores obtained during the bench mark experiment. As only POS tag for each word in the text was

added beyond the bench mark experiment, the improved performance score has resulted due to the POS tag information used during the experiment.

While the same POS tag information was applied to the corpus used for the bench mark experiment it has resulted with the decrease in performance score for SMO and Naïve Bayes algorithms but some improvement has been seen for bootstrapping algorithms. The performance score of Naïve Bayes algorithm was 75.32%, 67.26% for SMO, 68.7% for Bagging, 65.92% for ADtree and 65.47% for AdaboostM1.

The other experiment was to investigate the effect of involving POS tag information for unsupervised machine learning method and the performance score was 57.25% for EM and 56.07% for simple K-means. The performance score has resulted a decrease for EM unlike that of the bench mark but shows improvement for simple k-means.

As it could be inferred from the above experimental result for the three classes of machine learning paradigm, semi-supervised machine learning has been found to yield better accuracy with the inclusion of POS tag information.

In addition, the experiment has also been focused on investigating the effect of seed word for the five machine learning classification algorithms while POS tag information was used in the corpus. The result has shown that while POS tag information is used to the corpus effect of one seed word was encouraging for all of the classification algorithms unlike the experiment involving two or three. Unlike the bench mark experimental result, AdaboostM1, Bagging and SMO have scored better performance scores for one seed but ADtree and Naïve Bayes algorithm has been found to score better performance score using two seed word. This experimental result has shown that Naïve Bayes and ADtree algorithms show one seed word to yield better performance score while POS tag information is used during the experiment.

Lastly, the study was focused on the determination of optimal window size enough to be used for the development of WSD prototype model in Amharic language, as of this experimental result an average window size of 3-3 for Bagging, 4-4 for Naïve Bayes, 6-6 for ADtree and SMO and 7-7 for AdaboostM1 has been considered enough. The average optimal window size to be

used for Amahric language for WSD prototype development is hence 6-6. The bench mark experimental result for average optimal window size was 2-2 or 3-3 for the language the increase in window size might have been due to the increase in attributes due to the inclusion of POS tag for each word.

CHAPTER SIX

6. CONCLUSION AND RECOMMENDATION

6.1. CONCLUSION

Human Languages are the primary means used by people to communicate and record information. They have the potential to express an enormous range of ideas and convey complex thoughts. NLP deals with the study of natural language computations and is concerned with enabling computers to analyze and process natural languages. Even though development of tools and methods so far has concentrated on well-resourced languages, there has been greatest need for developing computational tools and applications for under-resourced languages as well.

Human languages involve multi-sense (polysemous) words, words that can be portrayed in different ways depending on the context in which they are used. Hence, natural language is fairly ambiguous so it requires a mechanism to be disambiguated. Ambiguity of words can be handled by human brain in a spontaneous way due to the capability of brain to get the correct sense from a given context but has been greatest challenge for computational linguists. The computational way of handling the ambiguity of words in context is called WSD and has been researched since the earliest days of MT during 1950's. WSD has become an open research area and concern in NLP due to its importance for many NLP tasks like MT, IR, speech and text processing etc.

WSD as one of the tasks involved NLP has been researched for the past few years particularly for well-resourced languages, of which one is English. Developing WSD systems has been described as an AI-complete problem, a problem that can be solved after resolving the difficult problems in AI such as representation of common-sense and encyclopedic knowledge. Even though many researches have been done on WSD for well-resourced languages, only initial research works have been done for Amharic language which all was focused on corpus based approach by incorporating a small size corpus collected by researchers.

Due to lack of standard linguistic resources (Dictionary, thesaurus, wordnet etc), previous research works on WSD for Amharic language have been carried out on corpus based approach unlike knowledge based or hybrid approaches. The corpus based approaches experimented for Amharic language were supervised, unsupervised, semi-supervised machine learning methods and ensemble of Naïve Bayes. Of the corpus based methods experimented so far, semi-supervised machine learning method have resulted encouraging performance score of 88.47% and has compromised knowledge acquisition bottleneck of supervised methods and lesser accuracy of unsupervised or ensemble methods.

Hence, this research work in overall has focused on investigating the effect of incorporating POS tag of each word in the dataset used in supervised, unsupervised and semi-supervised machine learning approaches. The study has used the data which has been collected, preprocessed and transliterated by previous researcher and has used 1031 Amharic sentences involving five ambiguous words.

Three experiments have been done in this study. The first experiment has focused on determining bench mark experimental result, the second was focused on building WSD prototype using POS tag information added on the already available corpus and lastly the third experiment has involved integration of POS tag information to the already available corpus beyond the second experiment. Clustering algorithms (EM and simple K-means) and five classification machine learning algorithms (AdaboostM1, ADtree, Bagging, SMO and Naïve Bayes) have been used during the experiment.

For having done the above experiment, a CRF POS tagger model developed for Amharic language by Solomon [84] has been used. In addition, clustering and classification machine learning algorithms which are available in weka-3.6.11 package have been used. Lastly, the experimental result of this study has been compared with previous research works done without the inclusion of POS tag information.

The experimental result of this study has shown that, effect of one seed word gives better performance score for all of the algorithms used in this study. The experimental result lies in the range of 80.79%-92.02818% for Bootstrapping and SMO algorithms and 60.6216% for Naïve

Bayes algorithm. POS tag information incorporated in this study have resulted improvement on the performance of the algorithms. The POS tag information has let ADtree and Naïve Bayes algorithms also to yield better accuracy using one seed word unlike the benchmark experimental result.

As the second experiment of this study was focused on investigating the effect of POS tag information on supervised learning, POS tag information has been attached on the already manually annotated dataset which was made ready by previous researcher. During the experiment, AdaboostM1 has resulted performance score of 65.47%, Bagging 68.73%, ADtree 65.92%, SMO 67.26% and Naïve Bayes 75.07%. The performance score for each of the three algorithms (AdaboostM1, Bagging and ADtree) has resulted improvement relative to the baseline score but that of SMO and Naïve Bayes have shown decrease in performance. The largest performance score of 75.07% has been resulted by Naïve Bayes algorithm. Hence, POS tag information attached to the fully annotated corpus has been seen to result performance degradation.

The third experiment has been carried out on investigating the effect of POS tag information on unsupervised learning methods using two clustering algorithms: EM and simple K-means. The experimental result has depicted performance score of 57.25% using EM and 56.07% using simple K-means algorithm. The performance of EM algorithm shows performance degradation as compared to 59.02% performance score of bench mark experiment. For simple k-means algorithm performance improvement has been seen as compared to 55.52% performance score of the bench mark experimental result.

The fourth experiment has focused on investigating the effect of POS tag information using semi-supervised machine learning approach, while POS tag information is attached after the data is fully labeled then it has resulted performance score of 89.32% for AdaboostM1, 79.90% for Bagging, 89.05% for ADtree, 86.80% for SMO and 46.71% using Naïve Bayes algorithm. This has resulted performance improvement as compared to bench mark experimental result except for SMO which has 87.4% score during the bench mark.

The other part of the fourth experiment has been done after POS tag information has been involved since clustering then the fully labeled data has been used for classification algorithms. During the experiment, the clustering task has been done using EM and Simple K-means algorithm then the fully labeled dataset has been used for classification purpose. The final experimental result has shown that semi-supervised learning apporach which has used fully labeled dataset resulted by EM to yield better performance score of 92.67% using ADtree, 92.33% using AdaboostM1, 89.93% using SMO, 80.98% using Bagging and 60.62% using Naïve Bayes algorithm. The experimental result obtained while fully labeled dataset is generated using simple K-means algorithm was 89.81% using ADtree, 87.87% using AdaboostM1, 87.26% using SMO, 83.47% using Bagging and 44.11% using Naïve Bayes algorithm. Hence, the experimental result shows better performance score could have been obtained while semi-supervised machine learning algorithm has been used on the POS tagged fully labeled dataset using EM algorithm.

The last experiment has been done on determining the optimal window size enough for developing WSD using Amharic language, the experiment has been done using POS tag information added on the fully labeled dataset and the other part was done after clustering is done for POS tagged information from the start. On the first part of the experiment incorporating POS tag information on the fully labeled dataset has not resulted any new result as compared to the bench mark experiment. During the second part of the experiment an increased window size of 3-3 using Bagging, 4-4 using Naïve Bayes, 6-6 using ADtree and SMO and an optimal window size of 7-7 has been obtained using AdaboostM1 algorithm. In general, the optimal average window size of 6-6 has been found out to be enough while POS tag information has been attached during the experiment involving Amharic corpus.

In general, incorporation of POS tag information on the corpus used for semi-supervised machine learning algorithm has been found to yield better performance score unlike supervised and unsupervised machine learning methods. The performance improvement while POS information has been added is: 4.2% for ADtree, 8.4% for AdaboostM1, 1.1% for Bagging, 2.5% for SMO and 12.6% for Naïve Bayes while seen in comparison with the baseline score. In

addition, it has been seen in this experiment that effect of one seed word to be better unlike that of two or three seed words. Lastly, it can be inferred that an optimal window size of 6-6 or 7-7 has been found to be enough for WSD using semi-supervised machine learning method using a corpus involving POS tag for each word of the text used during the experiment. The average optimal window size of 6-6 obtained in this experiment with the inclusion of POS tag goes in line with the former 3-3 window size obtained experimental results done by previous researchers as inclusion of POS tag information to each word will double the number of window size in the given text.

6.2 RECOMMENDATION

Based on the findings of this study and knowledge acquired from the previous works related to this study, the following recommendations have been forwarded for continual research works on the area to get better accuracy:

- As using larger corpus size deemed more realistic for WSD model built, extending this experimentation using semi-supervised learning for other ambiguous words in addition to those covered in this study is recommended.
- Due to the constraint of time, the researcher has experimented using two clustering and five classification algorithms so it is recommended if other combination of clustering algorithms and classification algorithms are experimented using POS tagged corpus.
- The focus of attention in this study has concentrated on using ambiguous Amharic words involving two senses. As there are ambiguous words in Amharic having three or more senses, further researcher could also result better performing WSD model to be built so it is recommended to be experimented using such Amharic words.
- This study has only concentrated on modeling WSD to tackle lexical ambiguity. Further researches are recommended if done using other sources of Amharic ambiguous words as well to address other type of ambiguities like semantic, phonological, referential etc.

REFERENCES

1. Abhishek Fulmari and Manoj B. Chandak (2013), A Survey on Supervised Learning for Word Sense Disambiguation, International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 12, December 2013

2. Gauri Dhopavkar, Manali Kshirsagar and Latesh Malik (2014), Handling Word Sense Disambiguation in Marathi Using a Rule Based Approach, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 International Conference on Industrial Automation and Computing (ICIAC)
3. Church, Kenneth W. and Lisa F. Rau(1995), Commercial Application of Natural Language Processing, In Communication of ACM 38:11 pp. 71-80

4. Bartosz Broda and Maciej Piasecki (2009),Semi-supervised Word Sense Disambiguation Based on Weakly Controlled Sense Induction, Proceedings of the International Multiconference on Computer Science and Information Technology pp. 17–24, ISBN 978-83-60810-22-4,ISSN 1896-7094

5. Getahun Amare (December,2001),Towards the Analysis of Ambiguity in Amharic, Institute of Ethiopian Studies, No. 2 , pp. 35-56
6. A. Farghaly and K. Shaalan (2009), Arabic natural language processing: Challenges and solutions, ACM Trans, Asian Lang. Inform. Process., 8:14:1–14:22
7. Thorsten Brants (2000), TnT- a statistical part-0f-speech tagger, In proceedings of the sixth Applied Natural Language Processing Conference ANLP-2000, April 29 - May3, 2000, Seattle, WA, Saarland University
8. Krister Linden (2005), Word Sense Discovery and Disambiguation, Helsinki University Press, ISBN 952-10-2472-0 (pdf),NO.37
9. Mohamed El Bachir Menai (2014), Word Sense Disambiguation using an evolutionary approach, Department of Computer Science, College of Computer and Information Sciences, Saudi Arabia, Riyadh 11543

10. Litkowski and K. C. (2005), Computational lexicons and dictionaries, Encyclopedia of Language and Linguistics (2nd ed.), K. R. Brown, Ed. Elsevier Publishers, Oxford, U.K.
11. Agirre E.; Lopez de Lacalle, A.; Soroa, A. (2009), "Knowledge-based WSD on Specific Domains: Performing better than Generic Supervised WSD" (http:/ / www. ijcai. org/ papers09/ Papers/ IJCAI09-251. pdf) *Proc. Of IJCAI*
12. Preeti Yadav and Sandeep Vishwakarma(May,2013), Mining association rules based approach to word sense disambiguation for Hindi Language, International Journal of Emerging Technology and Advanced Engineering Website, ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 3
13. Neetu Mishra, Shashi Yadav and Tanveer J. Siddiqui (2009), an unsupervised approach to Hindi Word sense Disambiguation, Allahabad, UP, india
14. Manish Sinha, Mahesh Kumar Reddy and Pushpak Bhattacharyya (2008), Hindi Word Sense Disambiguation, CSE Dept, IIT Bombay, Mumbai, India
15. Ide and V´eronis (1998),Introduction to the special issue on WSD: the state of the art, Computational Linguistics, pp1-40
16. Y. Wilks and M. Stevenson (1996), The grammar of sense: Is word sense tagging much more than part-of-speech tagging? Technical Report CS-96-05, University of Sheffield
17. Devendra Singh Chaplot (May,2014),Literature Survey on UnsupervisedWord Sense Disambiguation,Department of Computer Science and Engineering Indian Institute of Technology, Bombay
18. Ping Chen, Wei Ding, Max Choly and Chris Bowes (2011), Word Sense Disambiguation with Automatically Acquired Knowledge, Department of Computer and Mathematics Sciences, University of Houston-Downtown, 1 Main St., Houston,TX 77002. E-mail: chenp@uhd.edu
19. Taye Girma(May, 2014),International Journal of Advanced Research in Engineering and Applied Sciences, Human Language Technologies and Affan Oromo, ISSN: 2278-6252 Vol. 3,No. 5
20. Mukti Desai and Mrs. Kiran Bhowmick(October,2013), Word Sense Disambiguation, International Journal of Engineering Science Invention, ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726

21. Piek Vossen, David Farwell, German Rigau, Iñaki Alegria, Eneko Agirre and Manuel Fuetes (2006), Meaningful results for information retrieval in theMEANING project. Proceedings of the 3rd Global Wordnet Conference, Jeju Island, Korea
22. Mohd Shahid Husain and Mohd Rizwan Beg(2013), Word Sense Ambiguity : A survey, International Journal of Computer and Information Technology (IJCIT), ISSN:2279-0764, Volume 02- Issue 06
23. M.Humera Khanam, K.V. Madhumurthy and Md.A.Khudhus (september,2013), Part-of-speech Tagging for Urdu in scarce resource: Mix Maximum Entropy Modelling System,International Journal of Advanced Research in computer and communication Engineering vol.2, Issue 9
24. Varada Kolhatkar (August,2009), An Extended Analysis of a Method of All words Sense Disambiguation, Msc. thesis, University of Minnesota
25. Lorenza Moreno-Monteagudo, Ruben Izquierdo-Bevia, Patricio Martinez-Barco and Armando Suarez (2006), A study of the influence of POS tagging on WSD, Springer- Verlag Berlin Heidelberg LNAI 4188, pp. 173–179
26. Paul Lewis, Gary Simons and Charles Fennig (2013), Ethnologue: Languages of the World, Seventeenth edition,Dallas Texas: SIL International
27. L. Besacier, V-B. Le, C. Boitet, V. Berment (2006), ASR and Translation for Under-Resourced languages, Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 5:1221–1224
28. Wube Alemayehu (2005), Rule Based Syntactic disambiguation parser for Amharic language, Master's Thesis, Addis Ababa University, Faculty of Informatics
29. Teshome Kassie (2009), Word Sense disambiguation for Amharic text retrieval: a case study for legal documents. Master Thesis, Addis Ababa University
30. Solomon Mekonnen (2010), *Word Sense Disambiguation for Amharic Text, A Machine Learning Approach, Master's Thesis*, Addis Ababa University.
31. Solomon Assemu (June,2011), Unsupervised machine learning approach for word sense disambiguation to Amharic words, Master's Thesis, Addis Ababa University

32. Getahun Wassie (2012), Amharic WSD: Machine Learning Approaches', Master's Thesis, Addis Ababa University: School of Graduate Studies
33. Hagerie Woldie (June, 2013), Ensemble Classifiers Applied to Amharic Word Sense Disambiguation, Master's theis, Addis Ababa University, School of Information Science
34. Birhanie Ewunetu (Novermber, 2013), Ensemble of Naïve Bayesian Classifiers for Amharic Word Sense Disambiguation, Master's thesis, university of Gondor
35. Stevenson, M.; WILKS, Y. (2003), Word Sense Disambiguation in Mitkov (Editor) Oxford Handbook of Computational Linguistics, Oxford University Press, pages 249-265
36. Bar-Hillel, Yehoshua (1960). "Automatic Translation of Languages." In Alt, Franz; Booth, A. Donald and Meagher, R. E. (Eds), Advances in Computers, Academic Press, New York
37. Kilgarriff A. (1997), 'I don't believe in word senses', Computers and the Humanities,31(2), 91–113
38. Wilks, Y. (1975a), Formal Semantics of Natural Language, Preference Semantics, Cambridge University Press, 329-348.
39. Wilks, Y. (1975d), Proceedings of the workshop on Theoretical Issues in Natural Language Processing(Tinlap), association for Computational Linguistics, 38-41
40. Ravi Mante, Mahesh Kshirsagar and Dr. Prashant Chantur (2014), A Review of Literature on Word Sense Disambiguation, International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 5 (2) , 2014, 1475-1477, ISSN:0975-9646
41. E. Kelly & P. Stone (1975), Computer recognition of English word senses, in North-Holland Publishing Co., Amsterdam
42. S. Small & C. Rieger (1982), Parsing and comprehending with word experts (a theory and its realisation), in Strategies for Natural Language Processing. W.O. Lehnert &: M.H. Ringle, Eds., LEA: 89-148
43. M. Lesk (1988),"They said true things, but called them by wrong names" – vocabulary problems in retrieval systems, in Proc. 4th Annual Conference of the University of Waterloo Centre for the New OED.
44. Yarowsky, D. (1995), Unsupervised Word Sense Disambiguation Rivaling Supervised Methods, ACL-1995, pp. 189-196.

45. Brown, A. S. (1991), The tip of the tongue experience: A review and evaluation. Psychological Bulletin, 10, 204-223.
46. Sproat, Richard; Hirschberg, Julia; and Yarowsky, David (1992). "A corpus-based synthesizer." Proceedings of the International Conference on Spoken Language Processing, Banff, Alberta, Canada, October 1992.
47. Marine Carpuat and Dekai Wu (June,2007), Improving Statistical Machine Translation using Word Sense Disambiguation, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 61-72, Prague, Association of Computational Linguistics
48. Hwee Tou Ng (2011), "Does Word Sense Disambiguation Information Retrieval?", ESAIR, ACM 978-1-4503-0958-5/11/10, Glasgow, Scotland, UK
49. Sanderson (1994), Word sense disambiguation and information retrieval, Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 142–151.
50. M. Sundara Rajan (2010), A Study on effective information Retrieval Using Disambiguator Selector algorithm, Department of computer applications, PHD thesis,Chennai -600 095
51. Esha Palta(2006-2007), Word Sense Disambiguation, Master of Technology, Kanwal Rekhi School of Information Technology, Indian Institute of Technology, Powai, Mumbai
52. Snyder and Palmer (2004), The English all-words task In proceedings of the 3rd ACL workshop on the Evaluation of systems for the semantic Analysis of Text, Barcelona, Spain
53. Satanjeev Banerjee and Ted Pedersen(2003), Using measures of Semantic relatedness for word sense disambiguation, CICLing Proceedings of the 4th International Conference on Computational Linguistics and Intelligent text processing, Page 241-257, ISBN: 3-540-00532-3, Springer-Verlag Berlin, Heidelberg@2003
54. Roberto Navigli and Paola Velardi (2005), Structural Semantic interconnections: a Knowledge-based approach to Word Sense Disambiguation, Pattern Analysis and Machine Intelligence, IEEE Transactions on 27 (7), 1075 .
55. J. Sreedhar, S. Viswanadha Raju, A. Vinaya Babu, Amjan Shaik,P. Pavan Kumar.(2012), "Word Sense Disambiguation: An Empirical Survey", volume 2. IJSCE

56. Hal Daume (August,2012), A course in Machine Learning book
57. Mahesh Joshi (August,2006), Master's thesis in Kernel Methods for Word Sense Disambiguation and Abbreviation Expansion in the Medical Domain, university of Minnesota
58. Marquez, L.; Padro, L. & Rodriguez, H. (2000). A machine learning approach to POS tagging Machine Learning, 39, 59-91 Henrik Brink and Joseph W. (2014), Real-world Machine Learning, Manning publications co.
59. Leacock, C., Miller, G.A. & Chodorow, M.(1998), Using Corpus Statistics and Word Net Relations for Sense Identification, Computational Linguistics, 24:1, 147–165
60. Schutze, H. (1998), Automatic Word Sense Discrimination, Computational Linguistics, 24:1, 97–123.
61. Ms. Ankita Sati (2013), Semi-Supervised Learning Methods for Word Sense Disambiguation, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727, Volume 12, Issue 4
62. R. Mihalcea (2003), The Role of Non-Ambiguous Words in Natural Language Disambiguation, in Proceedings of the Fourth RANLP.
63. Shallu Shallu and Vishal Gupta (November,2013), A Survey of Word-sense Disambiguation Effective Techniques and Methods for Indian Languages,Journal of Emerging Technologies in Web Intelligence, Vol.5, No.4,Academy Publisher.
64. Schütze and Hinrich(1998), Automatic word sense discrimination,Computational Linguistics, 24(1):97–123.
65. Jiawei Han and Micheline Kamber(2006), Data Mining: Concepts and Techniques, 2nd edition, Morgan Kaufmann
66. chawla N.V. and Karakoulas G. (2005), Learning from labeled and unlabeled Data: An Empirical Study across techniques and Domains Journal of Artificial Intelligence and Research, Volume 23, Pages 331-366.
67. Pavan Kumar Mallapragada (2010), Some Contributions to Semi-Supervised Learning, M.S.Thesis, Michigan State University

68. Andres Montoyo, Armando Su´arez, German Rigau, Manuel Palomar (2005), Combining Knowledge- and Corpus-based Word-Sense-Disambiguation Methods, Journal of Artificial Intelligence Research
69. Avrim Blum and Tom Mitchell(1998), Combining Labeled and Unlabeled Data with Co-Training, proceedings of the 11th Annual Conference on Computational Learning Theory, pages 92-100
70. Etienne Grossmann, ADtree : boosting a weak classifier into a decision tree, ,University of Kentucky, Quality Street, Suite 857, Lexington, KY, 40507, USA.
71. Amanda Wimsatt and Rachel Wynn(2011), Amharic Language and Culture manual, Texas State University
72. Lewis, S. C., Zamith, R. and Hermida, A. (2013), Content Analysis in an Era of Big Data: A Hybrid Approach to Computational and Manual Methods. Journal of Broadcasting & Electronic Media, 57(1), 34–52. doi:10.1080/08838151.2012.76170
73. Gordon R. (2005), Ethnologue: Languages of the world, fifteenth ed. SIL International, Dallas.
74. Million Meshesha and C.V. Jawahar(June,2007), Optical Character Recognition of Amharic Documents, African Journal of Information and Communication Technology, Vol. 3, No. 2
75. Marvin L. Bender, Head W. Sydeny, and Roger Cowley (1976), The Ethiopian Writing System, In Bender et al (Eds.), Language in Ethiopia, London Oxford University press.
76. Baye Yemam (1987 ዓ.ም.), **የአማ ርኛ፣ሰዋስው። ት.መ.ማ.ማ.ድ. .።**
77. Yacob, D.(1996), System for Ethiopic Representation in ASCII (SERA), described to be cited from: http://www.abyssiniacybergateway.net/fidel/ on 12 March 2010
78. Dawkins, C.H (1969)., The Fundamentals of Amharic: A.A Sudan interior mission
79. Allen James (1995), Natural language Understanding, Redwood City: Benjamin/Cummings
80. R.L. Trask (1993), A dictionary of grammatical terms in linguistics, London and New York: Routledge, SSDI 0024-3841(94)00003-5
81. Dawit Bekele(2003),The Development and Dissemination of Ethiopic Standards and Software Localization for Ethiopia, The ICT Capacity Building Programme by Ministry of the FDRE and United Nations Economic Commission for Africa ,Addis Ababa.

82. Agirre E. and Martinez D. (2000),Exploring automatic word sense disambiguation with decision lists and the web. In Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content.
83. Martha Yifiru Tachbelie, Solomon Teferra Abate and Laurent Besacier (2011), Part-of-Speech Tagging for Under-Resourced and Morphologically Rich Languages-The Case of Amharic, Conference on Human Language Technology for Development, Alexandria, Egypt
84. John Laffterty, Andrew McCallum and Fernando C.N. Pereira(2001), Conditional Random Fields: Probablistic Models for Segmenting and Labeling Sequence Data, copyright ACM, pages 282-289
85. Sebastian Raschka(Oct 7,2014), An Overview of General Performance Metrics of Binary Classifier Systems
86. Ian H.Witten and Eibe Frank (2005), Data Mining: Practical Machine Learning Tools and Techniques. Second edition: Morgan Kaufmann publications.
87. Yoong Keok Lee , Hwee Tou Ng and Tee Kiah Chia(July,2004), Supervised Word Sense Disambiguatin with Support Vector Machines and Multiple Knowledge Sources,Third International workshop on the Evaluation of systems for the semantic Analysis of Text, Barcelona, Spain, Association for Computational Linguistics
88. Kaplan A. (1995), an experimental study of ambiguity and context, Mechanical Translation. , Vol. 2(2)
89. Roberto Navigli (February,2009), Word Sense Disambiguation: A survery, ACM ACM 0360-0300/2009/02-ART10
90. Andrew Brian Goldberg (2010), New Directions in Semi-supervised learning, A dissertation submitted for the degree of Doctor of Philosophy in computer science, university of Wisconsin-Madison
91. W. Gale, K. w. Church, D. Yarowsky (1992), Estimating upper and lower bounds on the performance of word-sense disambiguation programs, in Proceedings of the ACL, 30:249-256.
92. Bauer, E., & Kohavi, R. (1999), An Empirical Comparison of Voting Classification algorithms: Bagging, Boosting and Variants, Machine learning 36(1-2), 105-139

93. D. Jurafsky, Martin James (2007), Speech and Language Processing: An introduction to natural language processing Computational linguistics and speech recognition, Prentice-Hall
94. Sandipan Dandapat (2009), Part of Speech Tagging and Chunking with Maximum Entropy model, Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur India

Table of Contents

List of Figures

List of Tables

APPENDIX-A SAMPLE TRANSLITERATED DATA BEFORE POS Tagged

Rcontext6	Rcontext7	Rcontext8	Rcontext9	Rcontext10	class
TarA	baHsalaTana	Ala	Aynatocu	bAnko	say
lAy	HamErikA	mangHs	batawayAyu	yAqarabu	say
Hala	gl	flAgo	mArkA	TalA	say
zEgA	HasAlfo	saT	Hala	lEla	say
HqEndAywl	yAdargu	HqEnqfAto	mAla	Haydala	say
Salaf	Hala	baqrbu	ba.hgAwi	mangad	say
ymaTAlqEnA	Hayto	ymsl	Damar	hm	say
taHsartoHal	HahyAnA	HayA	Dbo	maSto	say
yTayqu	qomo	Haga	wadd	bil	say
Mdra	badA	waTA	yAhax	maTA	say
Hzb	kfA	wadd	tAwqAlahHqErsu	gbx	say
lAy	baHsarA	tankol	mASanaf	tacAlawHandu	say

APPENDIX-B

wordr7	tagr7	wordr8	tagr8	wordr9	tagr9	wordr10	tagr10	class
baHesalaTana	NP	HAla	N	HAyenatocu	N	bAneko	N	say
HamErikA	N	manegeHese	N	batawayAyu	VP	yAqarabu	VREL	say
gele	ADJ	felAgo	N	mArekA	N	TalA	N	say
HasAlefo	V	saTe	N	Hala	V	lEla	PREP	say
yAdaregu	VREL	HeneqefAto	VP	mAla	N	Hayedala	V	say
Hala	V	baqerebu	NP	bahegAwi	ADJP	manegade	N	say
Hayeto	V	yemesele	V	Damare	N	heme	N	say
HaheyAnA	NC	HayA	N	Debo	V	maSeto	V	say
qomo	V	Haga	N	wadede	N	bile	N	say
badA	NP	waTA	V	yAhaxe	VREL	maTA	N	say
kefA	N	wadede	N	tAweqAlaheHeresu	N	gebexe	N	say

APPENDIX-C SAMPLE OUTPUT OF CRF POS Tagger

sawu	sa	0	0	0	0	wu	N	
segA	se	0	0	0	0	gA	N	
naxA	na	0	0	0	0	xA	N	
bamaho	ba	0	0	0	0	ho	NP	
ganezabA	ga	ne	0	0	za	bA	N	
mAseqamaTe	mA	se	0	0	ma	Te	VN	
yate	ya	0	0	0	0	te	NP	
yAla	yA	0	0	0	0	la	PREP	
bAneke	bA	0	0	0	0	ke	N	
HAyena	HA	0	0	0	0	na	N	
Hala	Ha	0	0	0	0	la	V	
HabezA	Ha	0	0	0	0	zA	N	
Hanedi	Ha	0	0	0	0	di	N	
Hagare	Ha	0	0	0	0	re	N	
bAneke	bA	0	0	0	0	ke	N	
tabAla	ta	0	0	0	0	la	V	
TarA	Ta	0	0	0	0	rA	N	
baHesalaTana	ba	He	sa	la	Ta	na	NP	
HAla	HA	0	0	0	0	la	N	
HAyenatocu	HA	ye	0	0	to	cu	N	
bAneko	bA	0	0	0	0	ko	N	
1	0	0	0	0	0	0	NUMCR	
gelebaTA	ge	le	0	0	ba	TA	N	

APPENDIX-D SAMPLE OUTPUT OF WEKA-3.6.11 Tool

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	149	71.6346 %
Incorrectly Classified Instances	59	28.3654 %
Kappa statistic	0.4317	
Mean absolute error	0.3184	
Root mean squared error	0.4599	
Relative absolute error	63.7177 %	
Root relative squared error	92.0126 %	
Total Number of Instances	208	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.738	0.307	0.718	0.738	0.728	0.776	say
	0.693	0.262	0.714	0.693	0.704	0.776	live
Weighted Avg.	0.716	0.285	0.716	0.716	0.716	0.776	

=== Confusion Matrix ===

a	b	<-- classified as
79	28	a = say
31	70	b = live