2022-08

AMHARIC WORD SENSE DISAMBIGUATION USING TRANSFER LEARNING

NEIMA, MOSSA AHMED

BAHR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDENTS

FACULTY OF COMPUTING

MSC. FINAL THESIS ON

AMHARIC WORD SENSE DISAMBIGUATION USING TRANSFER LEARNING

BY

NEIMA MOSSA AHMED

AUGUST, 2022

BAHIR DAR, ETHIOPIA

AMHARIC WORD SENSE DISAMBIGUATION USING TRANSFER LEARNING

Neima Mossa Ahmed

A thesis submitted to Bahir Dar Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Information Technology in the Faculty of Computing.

Advisor: Million Meshesha (Ph.D.)

August, 2022
Bahir Dar, Ethiopia

DECLARATION

This is to certify that the title "**Amharic word sense disambiguation using transfer learning**". Submitted to the partial fulfillment of the requirement for Master of Science Degree in Information Technology under faculty of computing, Bahir Dar Institute of technology is recorded of original work carried out by me and has never been submitted to this or any other institution to get any other degree. The assistance and help I received during this investigation have been acknowledged.

Name of the student: ___________________ Signature_________________

Date of submission____________________

BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
FACULTY OF COMPUTING
Approval of thesis for defense result

I hereby confirm that the changes required by the examiners have been carried out and incorporated into the final thesis.

Name of Student Neima Mosso Signature ______ Date 8/17/2022

As members of the board of examiners, we examined this thesis entitled "**Amharic Word Sense Disambiguation Using Transfer Learning**" by: Neima Mossa Ahmed. We hereby certify that the thesis is accepted for fulfilling the requirements for the award of the degree of Masters of Science Degree in "**Information Technology**".

Board of Examiners

Name of Advisor	Signature	Date
Million Meshesha (Ph.D.)	million	8/17/2022
Name of External examiner	Signature	Date
Martha Yifiru (Ph.D.)		9/29/2022
Name of Internal examiner	Signature	Date
Abnew Ali		9/29/2022
Name of Chairperson	Signature	Date
Dagnachew Melsew		10/3/2022
Name of Chair Holder	Signature	Date
Abdulkerim Mohammed (Ph.D.)		05.10.2022
Name of faculty Dean	Signature	Date
Asegahegn E		05/10/2022

ACKNOWLEDGMENT

All thanks and gratitude are due to Allah (S.W.), without whose blessings, none of my work would have been possible.

I would like to thank my advisor **Dr. Million Meshesha** who helps me by giving critical comments, continuous help, suggestions as well as positive encouragement in the area.

I would like to express my gratitude to Bahir Dar University for providing this free scholarship without a requirement for additional research funds to encourage young and aspiring researchers.

I would like to thank Emebet Birhanu Hailu who is, the Injibara University Head of the Ethiopian languages and literature department, for supporting us by communicating with Amharic language doctors for dataset evaluation and annotation. I also to say thank you to Dr. Seid Muhie. for his help and support during dataset annotation and GPU support.

I would like to thank Dr. Tesfa Tegegn for his critical comments and advice starting from title selection.

I would like to thank my mother Hidja Yesuf and my brother Jibril Mussa for their unreserved love and comprehensive support. May Allah bless and long life for all my family.

Lastly, I would like to thank my friends for their support and advice.

ABSTRACT

Word sense disambiguation (WSD) plays an important role in different NLP applications such as information extraction, information retrieval, machine translation, and lexicography. The manual disambiguation process by humans is tedious, prone to errors, expensive, and time-consuming. Recent research in Amharic WSD used mostly handcrafted rules. Such works do not help to learn different representations of the target word (ambiguous word) from data automatically. Moreover, such a manual disambiguation approach looks at a limited length of surrounding words from the sentence. The main drawback of previous works is that the sense of the word will not be detected from the synset list unless the word is explicitly mentioned. Our study explores and designs the Amharic word sense disambiguation model by employing transformer-based contextual embeddings. More specifically, we have exploited the different operations provided by the transformer models, namely AmRoBERTa.

As there is no standard sense-tagged Amharic text dataset for the Amharic WSD task, we first compiled 800 ambiguous words from different sources, including the Amharic dictionary, Amharic textbooks (Grade 7-12), and the Abissinica online dictionary. Furthermore, we collect more than 33k sentences that contain those ambiguous words. The 33k sentences are used to finetune our transformer-based RoBERTa model (AmRoBERTa). We conduct two types of annotation for our WSD experiments. First, using linguistic experts, we annotate 10k sentences for 7 types of word relations (synonymy, hyponymy, hypernymy, meronomy, holonomy, toponymy, and homonymy). For the WSD disambiguation experiment, we first choose 10 target words and annotate a total of 1000 sentences with their correct sense using the WebAnno annotation tool. Each sentence with one target ambiguous word is annotated by two users and one curator (adjudicator). As preparing glosses for each sense is time taking, we prepare 100 glosses for the selected 10 targets.

We conduct two main experiments, word relationship classification using the CNN, Bi-LSTM, and BERT models and WDS disambiguation using the AmRoBERTa model with sentence similarity measures. For the classification task, the CNN, Bi-LSTM, and BERT-based classification models achieve an accuracy of 90%, 88%, and 93% respectively. For the WSD task, we have employed the FLAIR document embedding framework to embed the target sentences and glosses separately. We then compute the similarity of the target sentence with the glosses embedding. The gloss with the higher score disambiguates the target sentence. Our model was able to achieve an F1 score of 71%. Due to time constraints and the lack of Amharic WordNet, we could not experiment with a large number of training datasets. In the future, we plan to at least compile glosses for the 1000 sentences annotated using WebAnno and report the performance.

Keywords: Word Sense Disambiguation, Transfer Learning, Neural network, pre-trained Language Model, Natural Language Preprocessing, Morphological Analyzer, Amharic WSD.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AmSeg:	AmharicSegmenter
AmRoBERTa:	Amharic Robustly Bidirectional Encoder Representation from Transformer
BERT:	Bidirectional Encoder Representation from Transformer
BiLSTM:	Bidirectional Long Short-Term Memory
BNC:	British National Corpus
CNN:	Convolutional Neural Network
DL:	Deep learning
ELMO:	Embedding's from Language Models
GloVe:	Gloss vector
GPT:	Generative Pre-trained Transformer
LSTM:	Long Short-Term Memory
MT:	Machine Translation
NB:	Naïve Bayes
NLP:	Natural Language Processing
RQ:	Research Question
RNN:	Recurrent Neural Network
SVM:	Support Vector Machine
WSD:	Word Sense Disambiguation

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

1.1. Background of the study

Natural language processing (NLP) is a field of artificial intelligence that assists computers in understanding, interpreting, and manipulating human language. Natural language is now being used to exchange information among humans and has now reached the extent of being an evolution criterion for technology (Reta, 2015). To properly access and understand the information on the internet, there is a need for people all over the world to be able to use their language. This requires the existence of NLP applications such as machine translation, information retrieval, information extraction, and others. These downstream NLP applications rely on tools such as word sense disambiguation for their reasonable performance.

Most of the words in natural languages are polysemic, which means that they have several meanings (Hassen, 2015). Amharic is one of the languages that have many words with multiple meanings. It is like other Semitic languages with a morphologically complex structure (Senay, 2021). The ability to recognize what a word means from its context and solve ambiguity is one of the most difficult problems in natural language processing (Alian et al., 2016). Ambiguity is defined as a word, term, notation, sign, or symbol interpreted in more than one way (Mindaye et al., 2010). Word Sense Disambiguation is a central concern and a hard challenge in NLP, intending to determine the exact sense of an ambiguous word in a particular context (Huang et al., 2019). When WSD is used in conjunction with other NLP approaches, it improves the efficiency of identifying accurate keywords for use as features in classification, searching, and many more NLP application (Senay, 2021).

Based on the study of Ide and Vronis (1998), WSD tasks necessarily involve two steps. The first step is to determine all of the different senses for each relevant word (at least) to the text or discourse under consideration, i.e., to select a sense inventory from a list of senses in an everyday dictionary, synonyms in a thesaurus, or translations in a translation dictionary.

The second step is a method for assigning each occurrence of a word to the correct sense. In this step, the assignment of words to senses is accomplished by the dependency of two major sources of information namely the context of the word to be disambiguated as well as external and hand-devised knowledge sources. Unless a human explicitly provides the associations between word senses and context features in the form of rules, the computer will need to use machine learning techniques to infer the associations from some training material (Assemu, 2011).

Knowledge-based, as well as corpus-based, and hybrid machine learning methods are the main categories of approaches for WSD approaches (Pal & Saha, 2015). Knowledge-based WSD approaches are based on different knowledge sources such as machine-readable dictionaries (WordNet), thesauri, etc. LESK, semantic similarity, selection preference, and heuristic are the main algorithms. There are two sets of data for training and testing in supervised approaches. This approach to WSD systems employs machine learning techniques based on manually created sense-annotated data. The training set, which consists of examples related to the target word, could be used to learn a classifier. The supervised approach includes techniques such as Nave Bays, decision lists, and K-nearest neighbor algorithms. Unsupervised WSD methods do not rely on external knowledge sources, machine-readable dictionaries, or sense-annotated data sets, rather, they use the information found in unannotated corpora to differentiate the word meaning.

Recently, contextual embedding methods like BERT, ELMO, and GPT-2/3 learn sequence-level semantics by considering the sequence of all the words in the input sentence (Chawla et al., 2019). These methods are characterized by their high performance, and the ability to extract a lot of information from raw text. These recent language models, especially the BERT model is trained to predict the masked word(s) of the input sentence (El-razzaz et al., 2021). To weigh, the relationship between each word in the input sentence and the other words in the same sentence, BERT learns self-attention by giving a vector for each word. The vector represents the relationship of one word with other words in the input sentences and is used to generate word embedding.

The performance of BERT is extraordinary compared to ELMo, because, self-attention based transformer architecture is used, which, in combination with a masked language modeling target, allows to train the model to see all left and right contexts of a target word at the same time (Chawla et al., 2019).

1.2. Motivation

Word sense disambiguation is one of the difficult tasks in natural language processing (NLP) because, unlike humans, machines do not understand the meaning of ambiguous words from context. It is critical to developing a system that intelligently understands the contextual meaning of a word. Developing a WSD application is challenging, especially in languages with limited resources and morphologically rich languages like Amharic.

Word sense disambiguation is required in a variety of applications such as machine translation, information retrieval, and information extraction and it is regarded as an AI-complete problem (Alian et al., 2016). WSD plays an important role, in increasing the performance of NLP applications like machine translation, information retrieval, and text classification.

The study by Agirre and Edmonds (2006) explains the effect of WSD on the performance of different NLP applications in such a way that, the WSD component is a black box encompassing an explicit process of WSD that can be applied to any application, much like a part-of-speech tagger or a syntactic parser. The alternative is to include WSD as a specific "component" of the activity of a particular application in a specific domain and integrate it into a system completely that is difficult to separate (Agirre & Edmonds, 2006).

Kassie (2009), an Amharic language researcher, demonstrates how WSD can improve the effectiveness of an Amharic document query retrieval algorithm. The ability to find relevant documents is complicated because many words can have different meanings in different contexts. If search engines could disambiguate those words, more accurate retrieval of documents should be possible (Kassie, 2009). Mulugeta (2019) also indicated that for machine translation research, both lexical and structural ambiguities were challenges. Hence, we decided to build an automatic WSD solution that could consider contexts to disambiguate words that could have ambiguous meanings.

1.3. Statement of the problem

The Amharic language poses numerous challenges to WSD, because of its morphological richness and complexity. Several Amharic words have more possible senses for example the word **መንገድ** has many senses such as ጎዳና(street), ብልሀት(cleverness), ዘዴ(method), and አሰራር(procedure) and consequently makes the disambiguation task more difficult. The manual disambiguation process by humans is tedious, prone to errors, expensive, and time-consuming. To overcome such a problem, Amharic word sense disambiguation is mandatory.

Nowadays, there are different research works done for different languages, like English and Arabic by using BERT (Yap et al., 2020; Huang et al., 2019; El-razzaz et al., 2021). However, few researchers tried to use machine learning approaches (Hassen, 2015; Kassie, 2009; Wassie et al., 2014; Siraj, 2017; Mulugeta, 2019) and deep learning approaches (Senay, 2021) to develop Amharic WSD. These related works demonstrated efforts to develop Amharic WSD using various approaches and techniques, including supervised, semi-supervised, unsupervised, and knowledge-based with handcrafted features. Recent research used handcrafted rules or directly fetching the meaning of ambiguous words from the synset list, but did not learn different representations from data automatically. Even though the disambiguation looks at surrounding words from the sentence, the length of the context or the window size they used is small. WSD developed by researchers requires manually labeled sense examples for every word sense. So, if the sense of the word is not found in the synset list their system will not work. The previous study also use a small number of ambiguous words, they collected a maximum of 5 senses for one word.

Recent advancements in the NLP field showed that transfer learning helps in achieving state-of-the-art results for new tasks by using pre-trained models (Ezen-Can, 2020). Transfer learning, which employs pre-trained language models like BERT, has been demonstrated to be an effective machine learning approach for many natural language processing tasks (Agirre & Edmonds, 2006) such as word sense disambiguation (Chawla et al., 2019). It does not require defining features explicitly; instead, it aims to learn different representations from data automatically (Bouhriz et al., 2016). Hence, this study aims to design Amharic word sense disambiguation using a transfer learning approach that uses pre-trained transformer models.

Research Questions:

In an attempt to come up with Amharic WSD, this study explores and answers the following research questions.

RQ1: How to select and prepare the datasets required for constructing the WSD model?

RQ2: Which deep learning algorithm is suitable for constructing an optimal Amharic word sense disambiguation model?

RQ3: To what extent the proposed model works in disambiguating Amharic words with multiple senses?

1.4. Objective of the study

1.4.1. General objective

The general objective of this study is to explore and design Amharic word sense disambiguation model using a transfer learning approach.

1.4.2. Specific objectives

Based on the general objective, this study attempts to address the following specific objectives.

- To prepare a dataset for experimentation.
- To design architecture for Amharic Word Sense Disambiguation.

- To use transfer learning algorithms for the Amharic Word Sense Disambiguation.
- To develop a model for Amharic Word Sense Disambiguation.
- To test and evaluate the performance of the proposed model.

1.5. Scope and limitation of the study

Recently, the research discipline requires effective analysis of the Word Sense Disambiguation process. Transfer learning approaches include a pre-trained language model that can extract features from plain text. The focus of this study is to design and implement the disambiguation of Amharic ambiguous words using pre-trained models. Amharic word sense disambiguation has been done for sentence-level textual data collected from different data sources such as teaching material, Amharic news, Amharic Bible, Amharic dictionaries, and domain experts.

This study was limited to disambiguating words with their synonymy (a word with the same meaning), hyponymy (a sub-type of the word), hypernymy (supertype of the word), meronomy (part-whole relationship), holonomy (whole-part relationship), toponymy (cause and effect relationship), and homonymy (a word with the same punctuation and orthography) of word relation in a given context at the sentence level. Lexical, semantics, and orthographic, ambiguity is also included in our study. However, the rest of the phonological, structural, and referential ambiguities are not considered because of time, money, and data limitations.

1.6. Significance of the study

The purpose of designing WSD could be to improve the performance of various NLP applications such as machine translation, information retrieval (IR), information extraction (IE), Lexicography, and so on, allowing us to save time on the development of such tools and use them as an intermediate task. It can help with the development of the following NLP applications.

Machine translation (MT): WSD is required for MT because some words in each language have multiple translations with different meanings depending on the context (Sharma & Niranjan, 2015). These senses help to facilitate and correctly translate language within its context.

Information Extraction (IE) and Text Mining: In many applications, WSD is critical for accurate text analysis. To investigate the biological issue (Bioinformatics research) and Named Entity recognition system, information extraction is important. WSD is used as a preliminary step in these areas (Agirre & Edmonds, 2006).

Lexicography: WSD provides lexicographers with rough empirical sense groupings and statistically significant contextual indicators of sense, and lexicographers provide WSD with better sense inventories and sense-annotated corpora (Agirre & Edmonds, 2006).

Information Retrieval (IR): IR systems need to resolve the ambiguity of some queries to decide what information should be retrieved because of ambiguous words. Accurate disambiguation would enable it to eliminate documents that used the same words with different senses and retrieve documents that demonstrated the same meaning with different wordings. So WSD is important for query formulation and expansion.

In general, the findings of this research will provide experimental evidence demonstrating the use of AmRoBERTa for the development of an Amharic WSD model. The challenges and recommendations of this study will be important for Word sense disambiguation further studies. The resources from this study will be released to the GitHub repository to advance future research on WSD.

CHAPTER TWO

LITERATURE REVIEW

2.1. Overview

This chapter deals with the literature to design the proposed WSD model and we discuss the literature on the Amharic language, its writing system, ambiguity in Amharic, WSD approaches, and related works.

2.2. Amharic Language

Amharic is one of the northern Semitic languages of the Afro-Asian families, and it makes a significant contribution to literature from the 17th to the 19th centuries (ሃረገወይን፤ ፈትለወርቅከበደ፣ጽጌ., 1993). In Ethiopian language history, four major language groups have emerged: Cushitic, Omotic, Nilo-Saharan, and Semitic. After Arabic, Amharic (አማርኛ) is the second most widely spoken Semitic language (Gezmu et al., 2019).

It is the working language of the Federal Democratic Republic of Ethiopia and is also the working language of many regional states in the country like Amhara, Addis Ababa, South nations and nationalities, Benishangul Gumuz, and Gambella (Meckelburg, 2018). Besides, the language has a considerable number of speakers in all regional states of the country (Salawu & Aseres, 2015). Research done on Amharic has significant benefits because Amharic is not only spoken in Ethiopia, there are also speakers in Canada, the USA, Eritrea, and Sweden (Mulugeta, 2019).

2.3. Amharic writing system

Although the Amharic language was first used in Ethiopia in the 14th century, the language of literature at the time and until the 19th century was Geez, from which Amharic evolved. (Meshesha & Jawahar, 2008). The Amharic alphabet, known as ፊደል/Fidäl, was inherited from the Geez, an ancient South Semitic language, and is now only used by the Ethiopian Orthodox Tewahedo Church. ፊደል/Fidäl is a writing system in which consonants and vowels coexist within each graphic symbol. Unlike most Semitic scripts, such as Arabic and Hebrew, Amharic Fidäl is written from left to right. The writing system is made up of 231 core characters: 33 consonants, each of which has 7 "orders" depending on the vowel with which it is combined, and some additional orders of "ፊደል/ Fidäl are known as dikala hoheyat/ዲቃላ ሆሄያት (Getaneh, 2020).

There are no upper-case and lower-case latter variations and no conventional cursive (i.e. written in a connected letter) form in the Amharic writing system(Meshesha & Jawahar, 2008). The Ethiopic comma (፡) to separate words, Ethiopic full stop (።) to end the sentence, Ethiopic semicolon (፤) to separate Amharic words or phrases with similar concepts Ethiopic double dash (፥) to separate Amharic sentences with a similar concept and Ethiopic question mark (?) to end the question, is the main unique Ethiopic punctuation marks. They used to separate each word and sentence in a formal Amharic writing system. Nowadays, the Ethiopian modern writing system uses a single space rather than an Ethiopic comma (:) to separate words.

To improve the effectiveness and efficiency of the developed application, Natural Language Processing does not require the presence of punctuation marks, stop words, or different orthographic representations of the same meaning letters (normalization) in text processing (Mubatada’i et al., 2019). The same is true for Amharic language applications that delete various punctuation marks and common words, such as information retrieval

(IR), machine translation (MT), event extraction (EE), question answering (QA), and word sense disambiguation (WSD)(Senay, 2021).

Amharic is morphologically complex, so avoiding Ethiopic punctuation marks as well as content-bearing words would improve the efficiency and effectiveness of the automatic WSD model (Hassen, 2015). Despite the use of Amharic script ፊደል/Fidäl by speakers of the language, there are issues with standardization, such as the presence of "unnecessary" alphabets ፊደል/ Fidäl. Even though some Amharic ፊደል /Fidäl has the same pronunciation, they have various representations, for example, the ”Fidel” ሀ (ha) can have more than four representations (ሀ, ሐ, ኀ, ሃ), (Yimam et al., 2021) but the same phonemes to convey similar meaning in the language. For NLP processing, an arbitrary representation of words might pose a serious problem, example the word ሰው (man) and ሠው (man) might have different embeddings while it is the same word (Yimam et al., 2021).

2.4. Ambiguity in Amharic Language

According to Mindaye et al. (2010), ambiguity is defined as the property of being ambiguous, where a word, term, notation, sign, symbol, phrase, sentence, or any other form used for communication is interpreted in more than one way. Amare (2001) also defines ambiguity as, the quality of any thought, idea, statement, or claim whose meaning, or interpretation cannot be determined conclusively by a set of rules or processes. Specific and distinct interpretations are permitted in ambiguity, whereas it is difficult to form any interpretation at the desired level of specificity with vague information (Hassen, 2015b). According to Amare's (2001) research, the Amharic language contains six different types of ambiguities: lexical, phonological, structural, referential, semantic, and orthographic ambiguity.

2.4.1. Lexical Ambiguity

Because a word can have multiple meanings, different people will interpret the same word in different ways. When a lexical unit falls into two different part-of-speech categories with distinct senses, or when a lexical unit has more than one sense that all belong to the same part-of-speech category, lexical ambiguity is present(Amare, 2001) Its scope is on individual words or word-level understanding (Hassen, 2015; (Miangah & Khalafi, 2005). Under lexical ambiguity there are different causes such as categorical ambiguity, synonymy, homonymy, and homophone affixes (Wassie et al., 2014; Kassie, 2009; Assemu, 2011; Hassen, 2015; Mulugeta, 2019;) which are discussed below:

Categorical ambiguity:

Due to two lexical elements sharing the same phonological and homographic form but belonging to different word classes, this is the main reason for lexical ambiguity. The following ambiguous words in the sentence can be used to explain it:

Take the Amharic sentence "በቅሎ አየሁኝ'' and ''አክርማ ሰጠችኝ'' as an example. The word በቅሎ and አክርማ are ambiguous because they can be used as either a noun or verbal meaning. They have the following interpretation(Senay, 2021; Hassen, 2015):

1. በቅሎ አየሁኝ
   a. I saw a mule. When በቅሎ has a noun meaning.
   b. I saw something which is grown. When በቅሎ has a verbal meaning.
2. አክርማ ሰጠችኝ
   a. She gave me akirma (a kind of grass). When አክርማ has a noun meaning.
   b. She gave me something after delaying it for some time. When አክርማ has a verbal meaning.

Synonymy: Amharic words, such as “ካራ‟, “ሰንጢ” and “ቢላዋ" are synonymous, which means that their meanings are extremely similar. All of these words have the same function and the same senses which we call in English "knife".

Homonymy: are words that share a phonological structure yet have different meanings, creating ambiguity. In other words, it refers to words with similar lexical characteristics. These words may be ambiguous and pronounced and spelled similarly, but they have different meanings (Hassen, 2015),(E.Agirre, 2006). As Kassie (Kassie, 2009) described homonymy can be further divided into three types.

The first category is homographs, which are words with the same spelling but different sounds and meanings. The second type of homonymy is homophones, which are words that sound the same but have different spellings. For instance, air-heir and see-sea Thirdly, full homonyms are words that have the same pronunciation and spelling. Using a ball as an example, a ball is a gathering of people who dance.

Homophonous Affixes: When affixes are applied to different word classes, it happens. The prefix, the root, and the suffix are the three distinct morphemes that can be identified through morphological analysis of the word. Take the example sentence "ቤቱ ፈረሰ " as an example. This sentence can be read as follows:

1. ቤቱ ፈረሰ

   a. The house is destroyed.

   b. His house is destroyed.

From the above example “ቤት+ኡ ፈረሰ” the suffix /- u /can be used as a definite article or a third-person masculine identifier, so the sentence is ambiguous.

2.4.2. Phonological Ambiguity

There may be phonological ambiguity due to the placement of the pause in the word. When speakers pause or don't pause while speaking, a word becomes ambiguous or has multiple meanings (Kassie, 2009), (Mekonnen, 2010). As a result of variations in the placement of pauses within words, structures may be ambiguous.

The following two sentences are illustrating how phonological ambiguity arises (Mulugeta, 2019; Senay, 2021).

1) a. ደግ ሰው ነበረ= [dägg + säw] näbbär (he was a kind man.)

   b. ደግሰው ነበረ= [däggtsäw] näbbär (they had made a preparation for a banquet).

2) a. ስራ ስሩ ጥሩ ነው = [sira + siru] t’iru newi (it is good to work.)

   b. ስራስሩ ጥሩ ነው [sira siru] t'+ru näw (Various roots are good.)

In the above sentences, the pause (+) sign indicates where the pause is, when pronounced with pause there is a change in meaning.

2.4.3. Semantic Ambiguity

Concentrating on the interactions between word-level meanings in the sentence determines the potential meanings of a sentence. Semantic ambiguity is a result of polysemy, idioms, and metaphorical word relationships in sentences (Siraj, 2017),(Hassen, 2015).

Polysemy: Many ambiguous Amharic words can have different meanings by emphasizing certain characters while reading. The majority of Amharic words are polysemic, having multiple meanings. For instance, the meaning of "መብራቱ " is as follows:

1. መብራቱ ጠፋ

   a. The light went off.

   b. Mebratu (a person) disappears.

From the above sentence, the word መብራቱ may use as a definite noun phrase that is 'the light' or a person

Idioms: An idiom is a phrase that has a meaning other than the words' literal interpretations. Let's look at the following example and how they are interpreted:

1. ሁሉ አገርሽ
   a. every country is hers – is the literal meaning of the phrase ሁሉ አገርሽ.
   b. the idiomatic expression for ሁሉ አገርሽ is she is adaptable.
2. በሬ ወለደ
   a. The literal meaning for በሬ ወለደ is ab ox gave birth to a calf.
   b. an idiomatic expression is unheard of.

Metaphors: have literal or non-literal (metaphoric) senses. The following is an example of metaphoric ambiguity:

1. ቃል ሰጠ
   a. He makes conversation.
   b. He promised.
2. አራስ ነብር
   a. hot-tempered.
   b. leopard with newborn cubs.

2.4.4. Structural (syntactic) Ambiguity

By changing the word order and holding multiple potential positions or arrangements within the sentence's grammatical structure, syntactic ambiguity can convey more than one meaning. Rearranging the order at the syntactic level allows for the syntactic disambiguation of words that have multiple parts of speech (Assemu, 2011), (Kassie, 2009), (Hassen, 2015). For example, the sentence “የሀበሻ ታሪክ አስተማሪ”and “የጎጃም ገብስ ጠላ” have the following two different representations.

1. የሀበሻ ታሪክ አስተማሪ
   a. a person who teaches Abyssinian history.
   b. an Abyssinian who teaches history.
2. የጎጃም ገብስ ጠላ
   a. beer made of barley from Gojjam.
   b. beer of barley from Gojjam.

2.4.5. Referential Ambiguity

When a pronoun can be used to refer to more than one potential antecedent, ambiguity arises. Even if a pronoun is not written grammatically, it is still understood by default. For example, “ካሳ ስለተመረቀ ተደሰተ” this sentence has the following different readings (Kassie, 2009; Assemu, 2011; Senay, 2021):

1. ካሳ ስለተመረቀ ተደሰተ
   a. Kassa was pleased because he graduates.
   b. Somebody was pleased Kasa graduated.
   c. He was pleased because Kassa graduated.

2.4.6. Orthographic Ambiguity

Geminate and non-geminate sounds are causes of orthographic Ambiguity. This type of ambiguity can be solved using the context meaning of the sentence(Kassie, 2009)(Assemu, 2011), however, in some cases, orthographic ambiguity might not be possible to disambiguate. 'ልጁ ይስላል' and ገና are examples of orthographic ambiguity.

1. ልጁ ይስላል
   a. He draws ("yslal ")
   b. He coughs ("yslal ").
2. ገና
   a. yet
   b. Ethiopian festivals are celebrated once a year or at Christmas.

Therefore, the word "yislal" and "ገና" are the cause of orthographic ambiguity which have the same orthographic form for both the active and passive voice.

2.5. Overview of Word Sense Disambiguation (WSD)

Nearly all human languages have ambiguous words in Natural Language Processing (NLP), which are words whose sense varies depending on the context in which they are referenced. WSD is a fundamental task and long-standing challenge in Natural Language Processing (NLP), which aims to find the most proper sense for an ambiguous word in a particular context (Huang et al., 2016). WSD has proved to be a difficult problem and this is caused, at least in part, by the various types of sense distinction that occur in the language (Saeed et al., 2019). If a word can be understood in more than one way, each meaning being distinct, it is said to be ambiguous.

Any WSD task aims to allow the machines to understand the correct meaning of these ambiguous words like humans do. Based on Ide and Ide & Vronis's (1998) study, WSD tasks necessarily involve two steps. The first step is to determine all the different senses for each word that is relevant (at least) to the text or discourse in question. To do this, select a sense inventory from the lists of senses in an everyday dictionary, the synonyms in a thesaurus, or the translations in a translation dictionary.

The second step is a method for accurately and efficiently associating each occurrence of a word with its correct sense. The natural language community is still debating and grappling with the problem of precisely defining a sense(Ide & Vronis, 1998).

Word Sense Disambiguation is used in a variety of contexts, including text processing, speech recognition, information retrieval, machine translation, etc. Without WSD, the processing of data is error-prone (Majumder, 2021). Word sense disambiguation, when properly applied, has the potential to enhance NLP. The accuracy of the output that a system produces can be greatly improved by using WSD. In light of the context, it maintains the word's meaning.

2.6. Approaches for WSD

Today, depending on the knowledge type they use, different approaches are used to solve Word Sense Disambiguation problems. There are three different approaches for WSD; such as knowledge-based, corpus-based, and hybrid approach which is the combination of both corpus-based and knowledge-based, and contextual embedding without WordNet were different approaches used for solving WSD problems in different languages in the world. The procedures, knowledge sources, and algorithms they use are different for these three approaches.

2.6.1. Knowledge-Based Approach

Based on knowledge, try to distinguish between different words using the knowledge from the dictionary, WordNet, thesaurus, lexical database, thesauri, etc. to extract knowledge about the relationships between different words' senses. (Mulugeta, 2019)(Ghobadipasha, 2019). These techniques primarily aim to do away with the need for the massive amounts of training materials needed in supervised techniques(Bakx, 2006). As a result, the system can perform what is known as all-words disambiguation on words in running text. (Hassen, 2015). The main hindrance for use of the knowledge-based approach for an end to end applications was the lack of large-scale computational resources for evaluation, comparison, and exploitation with feasible costs(Reta, 2015). WSD can be further subcategorized into graph-based approaches and gloss-based approaches. There are typically four main categories of knowledge-based methods. These are LESK Algorithm (overlap-based approach), Semantic similarity, Selection preference(restriction), and Heuristic algorithms (Mulugeta, 2019).

LESK Algorithm: The Machine Readable Dictionary (MRD), on which LESK Algorithms are based, determines the overlap between the sense bag or context bag of two or more target words(Navigli, 2009). LESK algorithm is a very simple and old approach with less accuracy. It is the first-word WSD based on a machine-readable dictionary (MRD). The algorithm depended on the glosses of traditional dictionaries; these dictionaries often do not have enough words for this algorithm to work well. WordNet, which contains different types of relationships like synonymy, antonymy, toponymy, etc., can be used to overcome the LESK algorithm's shortcomings.

Selection preference: One knowledge-based strategy attempts to limit the number of context-relevant meanings of a target word by using selection preferences (Hassen, 2015). Using the knowledge source, selection preferences identify common sense and gather information about the likely relationships between word types (Ye & Baldwin, 2006) This method's fundamental premise is to count the instances of this type of syntactically related word pair in the corpus.

Semantic similarity: It is said that words that are related, share a common context and therefore the appropriate sense is chosen by those meanings, found within the smallest semantic distance(Ide & Vronis, 1998). Semantic similarity becomes extremely computationally intensive when more than two words are involved like LESK Algorithm.

Heuristic algorithms: Relying on heuristics derived from linguistic properties observed in large texts is a simple and relatively accurate method of predicting word meanings(E.Agirre, 2006). A heuristic is a technique that assigns senses based on three presumptions: the most frequent sense, one sense per discourse, and one sense per collocation(Mulugeta, 2019).

2.6.2. Corpus-based approach

The Corpus-based approach involves different ML(machine learning) techniques such as supervised, unsupervised, and semi-supervised techniques(Reta, 2015) to induce models of word usage from a huge collection of word examples. The learning phase, as well as the classification phase, are the two phases of the Corpus-based approach (Bakx, 2006). Learning a meaning classification model from the training examples makes up the learning phase. To assign the output meaning to new examples, this model is applied during the classification process.

Supervised corpus-based approach: these techniques outperform knowledge-based methods on all WSD evaluation datasets by learning the correct sense for each word from a sense-annotated corpus (Ghobadipasha, 2019). The supervised approach requires tagged corpora as a training set and always uses two sets of data for testing and training. Although manually creating tagged corpora costs money, it is highly effective. The knowledge acquisition bottleneck for supervised learning algorithms continues to present challenges

(Tadesse, 2021). The following section discusses the primary supervised algorithm that is employed for the task of word sense disambiguation:

Naïve Bayes: A Nave Bayes classifier is a straightforward probabilistic classifier that assigns a class of samples depending upon the application of Bayes' theorem (Siraj, 2017). During the training processes, the model probability is estimated using relative frequency. To counteract the impact of zero counts, a very straightforward smoothing technique has been used. Naive Bayes classifiers come in three main varieties: multinomial, Bernoulli, and gaussian. According to the Naive Bayes classifier,(Tadesse, 2021) Multinomial nave, Bayes is used for text and document classification as opposed to Gaussian and Bernoulli nave Bayes classifiers for continuous and boolean data, respectively.

Decision tree: based on Zhou's (Zhou & Han, 2005)study A tree structure that recursively divides the training data set is used to represent classification rules as a decision tree. A decision tree uses rules to divide the training dataset and choose senses. Each branch and internal node of a decision tree represent an output of the test that will be applied to a feature value. The sense of the word is represented for categorizing Boolean data when a leaf node is reached (Zhou & Han, 2005).

Semi-supervised corpus-based approach: there is a severe shortage of training data in NLP, so many Word Sense Disambiguation (WSD) algorithms use semi-supervised learning to address the problem by using both labeled and unlabeled data. Semi-supervised learning problems are those in which only a portion of the information in a large computer file (X) is labeled (Y). These issues lie somewhere between supervised learning and unsupervised learning (Mece et al., 2020). This field contains many universe machine learning problems. Because labeling data can be costly or time-consuming and might necessitate access to subject matter experts. Unlabeled data, however, is reasonable and easy to collect and store. Unsupervised learning methods can be used to identify and learn the structure of the input variables. Additionally, you can make the best guess predictions for the unlabeled data using supervised learning techniques, feed that data into the supervised learning algorithm as training data, and then use the model to make predictions for new, unused data (Amal & Ahmed, 2011).

Unsupervised corpus-based approach: According to Martn-Wanton and Berlanga-Llavori (2012), unsupervised WSD methods do not rely on external knowledge sources, sense inventories, machine-readable dictionaries (MDR), or sense-annotated data sets. Instead of assigning meaning to words, these algorithms typically discriminate between word meanings using data from unannotated corpora (Shirai & Nakamura, 2010). The unsupervised method is primarily used to get around this limitation in the supervised machine learning approach because gathering the necessary resources is a challenging and time-consuming process. Unsupervised techniques may therefore be used to overcome the knowledge acquisition bottleneck caused by the scarcity of extensive resources that have been manually annotated with word senses (Siraj, 2017; Navigli, 2009). The following discussion includes the main unsupervised methods for WSD.

Context clustering: the setting Using clustering techniques, the clustering method first creates context vectors, which are then grouped into clusters to determine the meaning of the word (Ranjan Pal & Saha, 2015), (Mulugeta, 2019)(Tadesse, 2021) a vector uses average agglomerative clustering to represent all word senses. The co-occurrence matrix is generated and similarity measures are applied, and then discrimination is carried out using any clustering technique.

Word clustering: This method is similar to context clustering, but instead of grouping context, it groups words that have the same semantic meaning. Based on Information Content (IC) on a single feature, the similarity is measured; the higher the Information Content, the more similar, and the lower the Information Content, the less similar(Tadesse, 2021). The final step in classifying the listed words into senses is to cluster them because they represent various uses of the word(Mulugeta, 2019). To do the clustering there is

Latent Semantic Analysis, Hyperspace Analogue to Language, and clustering By Committee algorithm (E.Agirre, 2006; Mulugeta, 2019).

Translational equivalence: This method of word sense disambiguation is multilingual because it calls for word-aligned parallel corpora in two different languages. A training context is made for a target word using its lexical or syntactic characteristics and its translation to the target language(Mulugeta, 2019).

2.6.3. Hybrid approach

A hybrid approach combines a corpus-based approach and a knowledge-based approach. The primary goal of this approach is to benefit from having more knowledge sources and the strength of different approaches(Mulugeta, 2019; Ranjan Pal & Saha, 2015; Tadesse, 2021).

2.7. Deep Learning

Deep Learning is a branch of machine learning and artificial intelligence that uses deep neural networks to perform significantly better on unstructured data. And became ubiquitous as a result of the expansion of high-performance computing facilities. (Mathew et al., 2021). These neural networks are inspired by and modeled after the human nervous system and brain anatomy to simulate the behavior of the human brain(Senay, 2021). Although removed from matching its ability allowing it to “learn” from large amounts of knowledge. Deep learning imitates the way humans gain certain sorts of knowledge. DL is a crucial component of knowledge science, which also includes statistics and predictive modeling.

Artificial neural network (ANN)-based deep (DL) learning technology has gained popularity as or hot topic in the computing world due to its capacity for learning from data. It is used extensively in a variety of application fields, including visual recognition, text analytics, cybersecurity, etc. (Sarker, 2021). Deep learning can be implemented using different architectures such as architectures like Unsupervised Pre-trained Networks, Convolutional Neural Networks(CNN), Recurrent Neural Networks(RNN), (Mathew et al., 2021), Long Short-Term Memory (LSTM), and Bidirectional Long Short-Term Memory (BiLSTM)(Mosavi & Ardabili, 2020)(Mathew et al., 2021).

2.7.1. Convolutional Neural Networks

In deep learning, a convolutional neural network (CNN, or ConvNet) could be a class of deep neural networks, most typically applied to image processing and NLP. Deep Learning's capacity to manage large amounts of knowledge over the past few decades has shown it to be a potent tool. Traditional methods are no longer as popular, especially when it comes to pattern recognition. Convolutional Neural Network (CNN) is one of the most well-known deep neural network types (Tsou et al., 2020). CNNs were first developed and used around the 1980s. At that time, CNN's primary capability was the recognition of handwritten digits. The most important thing to keep in mind about any deep learning model is that it requires a lot of computing power and a lot of information to train. This was a significant disadvantage at the time, which is why CCNNswere was restricted to the postal industry and avoided the globe of machine learning(Tsou et al., 2020).

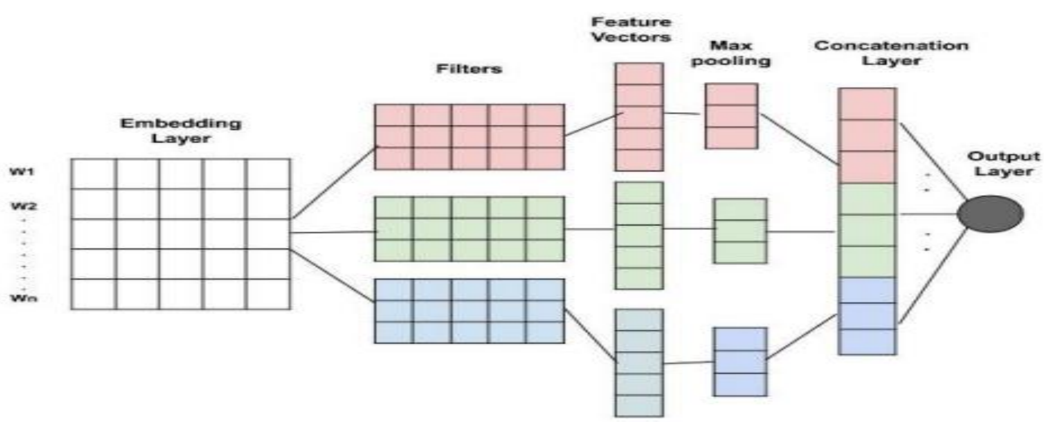


Figure 1:Conventional Neural Network(Ghosh et al., 2019)

2.7.2. Recurrent Neural Network (RNN)

A standard neural network has been made longer over time by using an RNN, which has edges that feed into the following time step rather than the following layer in the same time step(Mosavi et al., 2019). It is designed to recognize patterns and sequences in speech, handwriting, and text for use in various NLP applications. In recurrent neural networks (RNN), the current state is fed with the outputs from the previous states. CNN's hidden layers can retain information(Mathew et al., 2021).

Long short-term memory: A synthetic recurrent neural network (RNN) architecture called Long STM (LSTM) is used in the deep learning field. Given that there may be unknown time lags between significant events in an exceeding statistic, LSTM networks are well suited to classifying, processing, and making predictions supported by statistical data(Lewes, 2015). Similar to a recurrent neural network, an LSTM has a control flow. Because the information propagates forward, it processes the data and transmits it. The LSTM's cell operations are where the differences lie. These routine operations enable the LSTM to stay or forget information(Taneja et al., 2014).

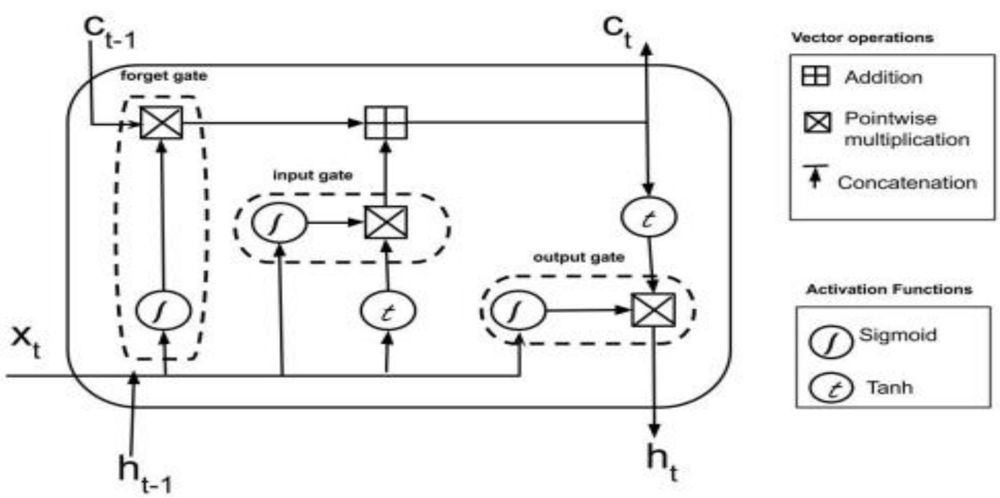


Figure 2: Long Short-Term Memory cell gate (Kleenankandy & K A, 2020)

Bidirectional LSTM: An extension of conventional LSTMs called a bidirectional LSTM may enhance model performance on sequence classification issues. Bidirectional LSTMs train two LSTMs rather than one on the input sequence in problems where all timestamps of the input sequence are available. A synthetic neural network is similar to how our brains are made up of thousands of neurons that work together to process information and respond to it automatically. In the world of programming, a neural network is a collection of algorithms that attempts to recognize underlying relationships in a set of data using a method that imitates how the human brain works. Neural networks can adapt to changing input, so there is no need to change the output criteria since the network generates the most straightforward result possible.

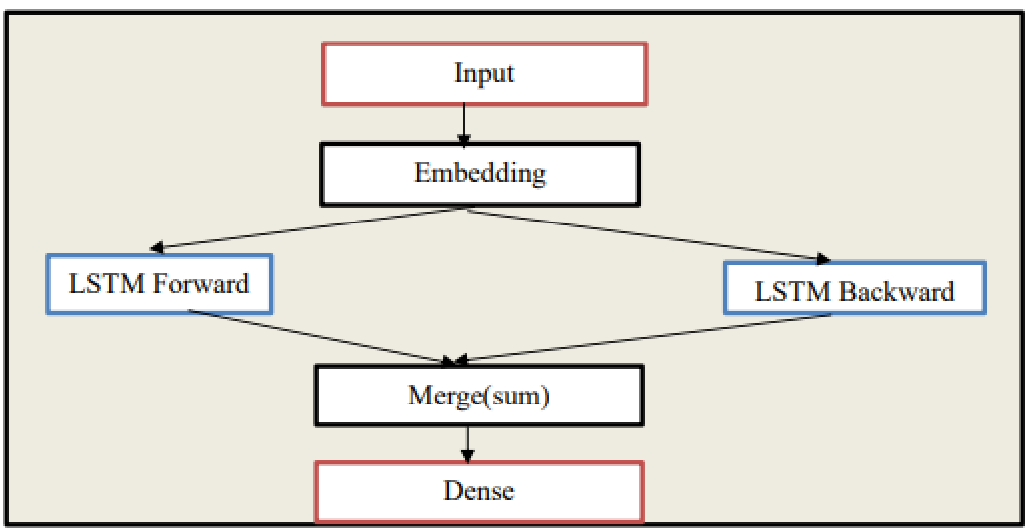


Figure 3: BiLSTM Architecture Diagram(Braz et al., 2018)

2.7.3. Unsupervised pre-trained Neural Networks

Over the last two years, the field of Natural Language Processing (NLP) has observed the rise of several transfer learning methods and architectures that significantly improved upon the state-of-the-art on a wide range of NLP tasks(Ruder et al., 2019). When previously acquired knowledge and skills have an impact on how new knowledge and skills are learned and applied, there has been a transfer of learning(Ruder et al., 2019).

Recent studies in learning contextualized word representations from language models, such as ELMo, BERT, and RoBERTa attempt to improve the issue of insufficient labeled data by first pre-training a language model on a large text corpus through self-supervised learning(Yap et al., 2019). Transfer learning aids us by creating a strategy to use the knowledge gained from one or more source tasks to enhance learning in a related target task, as shown in figure 4. Many natural language processing tasks have been shown to benefit from domain adaptation or transfer learning using pre-trained language models, such as BERT.

According to recent findings, deep neural networks that use contextual embeddings perform significantly better than those that don't on the majority of text classification tasks(Sikonja, 2018). Contextualized word representations are effective in downstream natural language processing tasks like question answering, named entity recognition, sentiment analysis, and word sense disambiguation. They can provide various representations for the same word in various contexts. (Hadiwinoto et al., 2018).

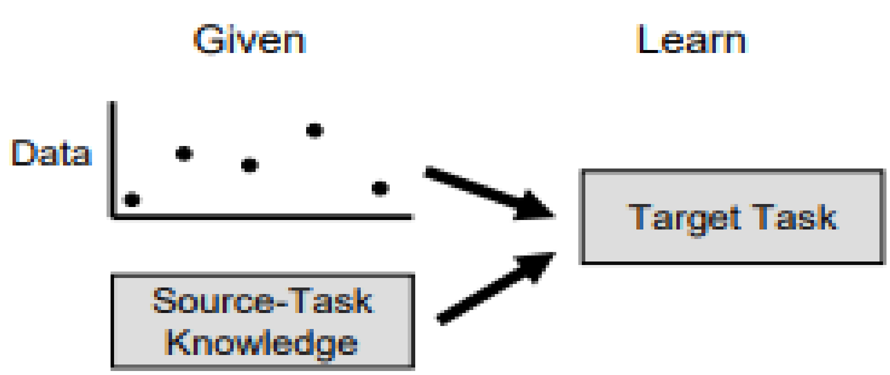


Figure 4: Transfer Learning(Aggarwal, 2014)

Embedding from language model (ELMo): is one of the pre-trained transfer learning models that solve word embedding models or representation problems by introducing contextual components. Traditional word embeddings such as word2vec, Glove, and FastText do not capture the context, though, so each word is always given the same vector, regardless of its context or meaning(Sikonja, 2018). Based on Sikonja's (Sikonja, 2018) study ELMo model's architecture consists of three neural network layers. The first layer is a CNN layer, which operates on a character level and is context-independent, so each word always gets the same embedding, regardless of its context. It is followed by two MLM layers, that consist of two concatenated LSTMs. Since ELMo is trained on a bidirectional network to predict the n-th word, it considers both the words before the n-th word and the ones after it(Ghobadipasha, 2019).

Bidirectional Encoder Representations from Transformer (BERT): (Devlin, 2018)(Hadiwinoto et al., 2018) BERT is an unsupervised re-trained language model, which is a fully trained language model by google. It is pre-trained using two unsupervised tasks like next sentence prediction and masked language modeling. By jointly conditioning both left and right context in all layers and using unlabeled text, BERT is designed to pre-train deep bidirectional representations. It has a clear conceptual foundation and strong empirical support. (Devlin, 2018). BERT has been adopted by various state-of-the-art models such as RoBERTa.

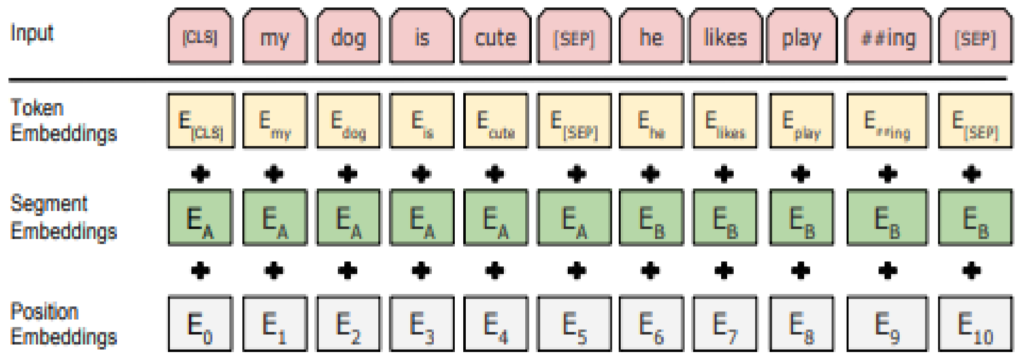


Figure 5: BERT input representation from (Devlin, 2018)

A Robustly Optimized BERT Pretraining Approach (RoBERTa): One of the most exciting architectures, RoBERTa, a copy of BERT created by Facebook, is one that was developed. In contrast to BERT, RoBERTa did away with the "next sentence prediction" functionality to train on longer sequences and dynamically change the masking patterns(Yimam et al., 2021). Based on Yimam's (2021) study Models based on contextual embeddings from RoBERTA perform better than static embeddings like word2Vec models(Yimam et al., 2021).

2.8. Related works

Some descriptions of foreign and local related works in terms of (problem attempted, approaches followed, results achieved, and, the way forward suggested) are provided as follows.

2.8.1 Word Sense Disambiguation for foreign language

Marwah and Arafat (2016) tried to demonstrate their study by using Wikipedia for Arabic Word Sense Disambiguation. Their study was looking for developing a new approach for Arabic Word Sense Disambiguation using the knowledge-based approach, where the text is preprocessed and the senses of the ambiguous words are retrieved from Wikipedia. After the retrieved senses, the tested text is represented as vectors where the cosine for the angle between the two vectors is computed. For evaluating the proposed approach, they conducted three different experiments. In the first experiment, they compared the raw frequency VSM of the text containing the ambiguous word with one sentence of Wiki text containing the word. In the second experiment, they used one retrieved sentence from Wikipedia but the retrieved texts and the actual text are represented using a Tf-Idf vector space model. In the third experiment, the extracted text from Wikipedia is expanded to be one paragraph containing the ambiguous word, and the Tf-Idf vector space model is applied. The third experiment gives the correct meaning of the ambiguous word. To produce a better result, the local context of the ambiguous word will be considered with the word's global context as their recommendation.

Based on Bouhriz et al. (2016) The majority of Arabic word-separation systems (WSDs) rely on data extracted from the word's local context. Typically, the best disambiguation cannot be made with this information. They offer a method to get around this limitation that extracts the global context from the full text in addition to the local context. By combining local context with a global context, they tried to solve Word Sense Disambiguation for Arabic text. Their experimental result showed that the proposed system has state of an art results with an accuracy of 74%.

at el. (2019) attempts to develop English Word sense disambiguation using BERT. They proposed to use BERT to extract better polyseme representations for WSD and explore several ways of combining BERT and the classifier. Sense definitions are utilized to train a unified classifier for all words, which enables the model to disambiguate unseen polysemes. Even though, the framework provides two annotated corpora for training: Semcor and OMSTI. they choose SemCor as the training dataset. Experiments show that the proposed model achieved state-of-the-art results on the standard English All-word WSD evaluation. In future works, they recommended using the relations between senses, like hypernym and hyponym, to provide more accurate sense representations.

Based on Chawla at el. (2019) study there are recent innovations in natural language processing such as ELMo, Flair NLP and BERT called Contextualized word embeddings (CWE). They introduce a simple but effective approach to word sense disambiguation using a nearest neighbor classification on CWEs. Contextually embedded token representations are advantageous compared to static word embeddings for several tasks such as text classification and sequence tagging. They conducted two experiments to determine whether contextualized word embeddings can solve the WSD task. To compare different CWE approaches, they used k = 1 nearest neighbor classification, simple kNN with ELMo as well as BERT embeddings. In their experiment, BERT achieved a state-of-the-art result.

The study by Huang et al. (2019) stated that word sense disambiguation is the means of finding the exact sense of an ambiguous word. The study focuses on the way to improve the use of gloss knowledge in a supervised neural WSD system. They build context-gloss pairs and propose three BERT-based WSD models. They improve the pre-trained BERT model and attain new state-of-the-art results for the WSD task results. In this research, they aim to improve the use of gloss knowledge in a supervised neural WSD system. By building context-gloss pairings and subsequently turning WSD into a sentence-pair classification task, they suggest a new approach to WSD. Our approach for Amharic WSD is similar to the approach by Huang et al. (2019) except that we exploit the sentence similarity approach rather than building sentence classification models.

EL-Razzaz's (2021) study states that WSD aims to predict the correct sense of a word given its context. Arabic written words are highly ambiguous and to solve this ambiguity they present an Arabic gloss-based WSD technique. In this study, they utilize the celebrated Bidirectional Encoder Representation from Transformers (BERT) to build two models AraBERTv2, and ALBERT, that can efficiently perform Arabic WSD. They divide the data set into three equal portions for training, validation, and testing to evaluate the performance of the proposed models: 60%, 20%, and 20%, respectively. Their experimental result showed that the new models outperform recent gloss-based WSD systems because they used a pre-trained BERT model.

2.8.2. Word Sense Disambiguation for local language

Kassie (2009) tried to demonstrate WSD for Amharic language using semantic vector analysis. A total of 865 words were selected from the Ethiopian Amharic language legal statute documents. Instead of using sense-tagged words, the researcher evaluates WSD using pseudo-code words (artificial words). The developed algorithm outperformed the one used by Lucene, according to their comparison of the two. The achieved result is an average precision and recall of 58% and 82%, respectively. The author recommended developing resources such as Corpora, Thesaurus, and WordNet, that could be useful advance the research in information retrieval, and word sense disambiguation.

Mekonnen (2010) conducted the Amharic WSD study using a corpus-based, supervised machine-learning approach. The author used the Naïve Bayes algorithm for Amharic WSD to classify a word to its correct sense using Weka 3.62 package in both the training and testing phases. These techniques, however, struggle with the issue of a knowledge acquisition bottleneck, in which the classifiers are only given a finite amount of labeled data. A total of 1045 English sense examples for the five ambiguous words were gathered from the British National Corpus (BNC). The dictionary is used to translate the sense illustrations back into Amharic. For each sense of the ambiguous word, a total of 100 sentences were collected; the accuracy achieved ranged from 70% to 83.5% for all classifiers.

Assemu (2011) tries to develop corpus-based Amharic WSD through the use of unsupervised machine learning. A total of 1045 English sense examples for the five ambiguous words were gathered from the British National Corpus (BNC). Using the Amharic-English dictionary, the sense examples were converted to Amharic and prepared for experimentation. It was done using unannotated training data that contained the target word in it. The authors used the current Weka 3.6.4 package implementation to test five clustering algorithms (simple k means, hierarchical agglomerative: Single, Average and complete link, and Expectation-Maximization algorithms). The result showed that the accuracy of unsupervised Amharic WSD is state-of-the-art result than the supervised machine learning approach, with an accuracy of 83.2% and 70.1%, respectively. For better Amharic WSD, the researcher recommended using linguistic tools like the Thesaurus, Lexicon from WordNet, machine-readable dictionaries, and machine translation tools.

Wassie (2014) utilized a semi-supervised learning strategy, and present a WSD prototype model for Amharic words. Unsupervised machine learning approach for clustering based on instance similarity and supervised machine learning approach after unlabeled data are applied. To cover all the senses of each target word available to sense annotated corpora are highly insufficient. The development of the Adaboost Bagging and ADtree algorithms perform at 84.90%, 81.25%, and 88.45 %, respectively. The author concludes that Semi-supervised learning using bootstrapping algorithm performs better.

The research by Hassen (2015) to extract knowledge from word definitions and relationships between words and senses, an Amharic WSD knowledge-based approach based on WordNet was developed. They manually created the Amharic WordNet for this study and chose 2000 words, including ambiguous words. They carried out two tests to compare Amharic WordNet's impact with and without a morphological analyzer, and the results showed an accuracy of 57.5% and 80 %, respectively. A two-word window on either side of the ambiguous word is sufficient for Amharic WSD, according to their research into the optimal window size. In this experiment, they have concluded that Amharic WordNet with a morphological analyzer can have better accuracy than without a morphological analyzer. They recommended automatic WordNet development and a hybrid approach.

(Tesema et al., 2016) To automatically gather disambiguation information, the researcher applied supervised machine learning techniques to a corpus of Afaan Oromo language. This method is known as a corpus-based approach to disambiguation. To determine the prior probability and likelihood ratio of the sense in the provided context, it additionally utilized the Naive Bayes approach. A total of 1240 Afaan Oromo sense examples were gathered for the chosen five ambiguous words, and the sense examples were manually tagged with their appropriate senses. The author used a corpus of Afaan Oromo sentences based on the five selected ambiguous words to acquire disambiguation information automatically. The co-occurrence feature, which indicates word recurrence within a certain number of words to the left or right of the ambiguous word, and the k-fold cross-validation statistical technique were the contextual characteristics used in this work. However, the supervised machine learning approach of the WSD performs better with human intervention; however, this research has limitations of the knowledge-acquisition bottleneck, i.e., it requires manually labeled sense examples which take a lot of time, are very laborious, and are very expensive to create when the corpus size increases and training data is required. The researcher's accuracy rate was 79%, and she discovered that the Afaan Oromo WSD can handle four words on either side of an ambiguous

Siraj (2017) attempts to develop a system for word WSD that uses data from Word-Net and tagged example sentences to determine the sense of ambiguous Amharic words. Information from WordNet was extracted using the LESK algorithm and Python programming. The WordNet is made up of 17 ambiguous words from various classes, along with developed synonyms and glossary definitions. Based solely on the Jaccard Coefficient and Cosine Similarity, Amharic WSD's accuracy performance reached 84.52% percent and 85.96%, respectively. The average accuracy of the Jaccard Coefficient with Lesk scores is 89.83% which is a better result, compared to Cosine similarity with LESK (86.69%). The researcher suggests for future work to use the Adaptive LESK algorithm and improve the performance of the WSD system.

Mulugeta (2019) attempts to develop an Amharic WSD system that uses Amharic WordNet hierarchy as a knowledge base. They use context to gloss overlap augmented semantic space approach. Most previous research on Amharic WSD focused on verb class; yet, Mulugeta (Mulugeta, 2019) tried to solve all open classes (verb, noun, adverb, and adjective) by developing WordNet. The WordNet contains about 250 synsets and does not include all relationships for single-sense words in the WordNet. The main challenge in this study was the unavailability of lexicon resources (WordNet), and the stemmer algorithm used in the preprocessing does not cover all exceptions and has limitations in returning the root word. Experimental result shows that context-to-gloss followed by augmented semantic space has achieved the highest recall 87% and 79% for three target words at word and sentence level respectively. And the highest average accuracy, 80% and 75% at word-level and sentence level are achieved by this approach. Their recommendation is to develop a better stemmer or morphological analyzer and fully constructed WordNet containing relationships for non-ambiguous words.

(Tadesse, 2021) In this research, A machine learning-based word sense disambiguation model for the Wolaita language was proposed. A total of 2797 sense instances were gathered to complete the investigation. Language specialists assessed the acquired data before creating five datasets for five ambiguous words, including "Doona," "Ayfiya," "Aadhdha," "Naaga," and "Ogiya. They used quantitative and experimental research to discover the ideal machine combination algorithms for learning and methods for extracting features. AdaBoost classifier utilizing BOW, TF-IDF, and Wor2Vec feature extraction approaches, Support Vector Classifier, Bagging, Random Forest Classifier, and AdaBoost classifier was chosen and trained using five datasets. In this study, precision and recall were used as the primary metrics for evaluation. Support Vector Classifier and Bagging classifiers with TF-IDF obtain an accuracy of 83.22% and 82.82%, respectively.

Recently, Senay (2021) tries to develop Amharic WSD by using a deep-learning approach. A total of 159 ambiguous words, 1214 synsets, and 2164 sentence datasets were used to create three distinct deep learning algorithms in three separate experiments. As a methodology, they used a design science research strategy. The author used different deep learning models for classification such as (LSTM, CNN, and Bi-LSTM) that trained on the dataset using hyperparameters. The results showed that LSTM, CNN, and Bi-LSTM obtained 94%, 95%, and 96% accuracy during the third experiment, respectively. But for disambiguation, they used handcrafted rules without applying any model. To increase the performance of the model, using lemmatization in the preprocessing, and using an attention mechanism are recommended.

Generally, Amharic word sense disambiguation was done by different researchers using different machine learning approaches. However, there is no easy and automatic Amharic word sense disambiguation, and there is no research that used the Transfer learning algorithm for the disambiguation purpose. Generally, most of the literature tries to develop Amharic WSD but there is a gap in solving the problems of word sense. Most of them follow a manual approach for extracting word sense. Recent research used handcrafted rules or directly fetching the meaning of an ambiguous word from the synset list or in the WordNet, but did not learn different representations from data automatically. Even though the disambiguation looks at surrounding words from the sentence, the length of the context or the window size they used is small. WSD developed by researchers requires manually labeled sense examples for every word sense. So, if the sense of the word is not found in the synset list their system is working. Previous studies consider simple and compound sentences during dataset preparation. The previous study also used a small number of ambiguous words and they collected a maximum of 5 senses only for one word.

Previous researches also require defining features explicitly; but transfer learning algorithms aim to learn different representations from data automatically (Bouhriz et al., 2016); solve ambiguity problem based on sentence semantics. In this research, we attempt to employ Transfer Learning for Amharic WSD.

CHAPTER THREE

METHODOLOGY

3.1. Overview

This chapter deals more with the design part of the proposed WSD model and we discuss different text pre-processing tasks like tokenization, normalization, stop word removal, and morphological analysis. To assess the model performance, we also deal with the model training process, performance measures, and data set.

3.2. Research design

A research methodology is an action plan, strategy, process, or design that lies behind the selection of methods and links the selection of methods to their use (Desmond Bala Bisandu, 2019).

In this study, we employ experimental research methodologies, which are widely used in computer science. The reason for selecting this research methodology is to select the optimal model for the proposed work. So, applying different experiments is used to select a more suitable or optimal model for our research. This research design helps us for fixing Amharic document ambiguity. Using this research methodology different experimental setups were implemented and evaluated their effect on the proposed research work. The proposed work passes the following phases: literature review, data collection and preparation, dataset annotation, preprocessing, feature extraction, model construction, and evaluation.

3.2.1 Literature Review

We conducted a literature review and state-of-the-art solutions to better understand the problem domain and identify gaps. In this study, we analyze and evaluate various works of literature from thesis reports, journals, conference proceedings, the Internet, and books to gain knowledge about the recent state-of-the-art problem. In addition, we conduct a literature review to gain a clear understanding of the various approaches to WSD from related publications such as thesis reports, journal articles, conference papers, and the Internet. The literature review will aid in understanding the problem domain, identifying gaps, and preparing a dataset for experimentation. In addition, discussions with domain experts were held to identify the problem and the cause of the problem with motivation.

3.2.2 Data collection and preparation

Since there are no labeled datasets available for Amharic word sense disambiguation, the main task for this thesis work is to prepare labeled datasets for WSD. The dataset is gathered from a variety of sources, including Amharic dictionaries, Amharic textbooks, the Amharic Bible, Amharic news, and Amharic Quran.

The collected data passes through data preprocessing in an attempt to prepare the data for experimentation. Data preprocessing is critical for improving the performance of the model. To make our data more suitable for the experiment, we use various data preprocessing techniques such as tokenization, stop word removal, special character removal, normalization, and then morphological analysis before data manipulation. Data must be preprocessed before it can be used for training or testing.

3.2.3 Dataset annotation

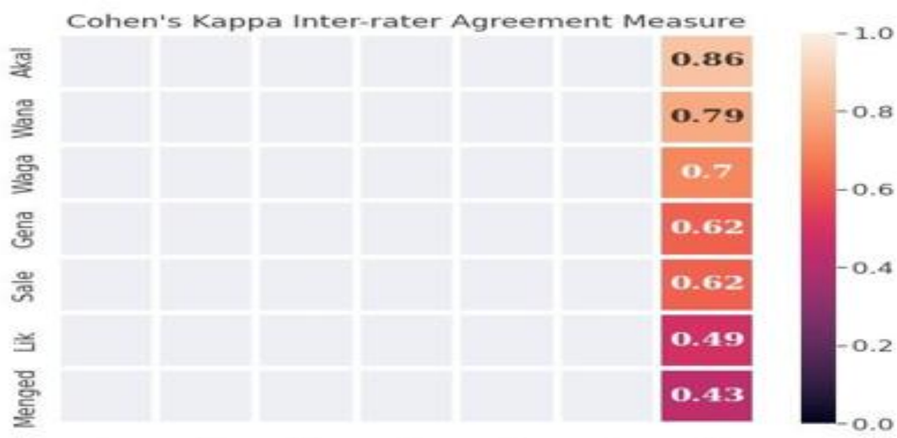
In our study, we selected annotators to keep the nature and behaviors of Amharic language texts, and to acquire quality and reliable data we tried to annotate both relationship of the sentence and the sense of the word in the sentence.

For the dataset annotation, we have done two different annotations. The first annotation is, to know whether the data set contains all the selected relationships of a word or not. We prepared a dataset annotation guideline to avoid bias and subjectivity (attached in appendix B). Therefore, we selected three Amharic language and Literature Department experts to annotate the data. Two annotators may disagree during annotation, so the third annotator is needed for a decision. The experts annotated the relationship between the sentence in the sentences. The inter-annotation agreement for word relation is shown in table 1.

The second annotation is for disambiguation or to know the sense of the word. For this task, we have also used the WebAnno annotation tool to annotate the ambiguous word in the sentence. As shown in appendix C we selected two annotators and one curator from Amharic language native speakers. The annotators annotate the sense of the word in the sentence by using the WebAnno annotation tool. From this annotation tool, we have obtained a better advantage over the static or manual annotation method, because it helps facilitate the annotation. This tool is important for the next researchers, to easily get the annotated dataset. The main advantage of the WebAnno annotation tool is getting the value for inter annotation agreement (such as Fleiss kappa, and Cohen’s kappa) is easy. We used Cohen's kappa as a measure of inter-annotator agreement because we selected two annotators and a curator. The curator is used as a decision maker if two annotators disagree. We demonstrate the prototype by explaining each module, running it using sample data, and showing the result obtained.

Table 1:Inter-annotation Agreement for word relations

SUM AND AVERAGE RESULT			
Synonym	461	AVG	0.886538
Homonymy	459	AVG	0.882692
Meronomy	451	AVG	0.867308
Holonym	416	AVG	0.8
Hyperym	444	AVG	0.853846
Hyponym	443	AVG	0.851923
Toponymy	451	AVG	0.867308
Single sense	444	AVG	0.853846
All Agreed	219	AVG	0.421154



values ≤ 0	no agreement
0.01-0.20	as none to slight,
0.21-0.40	as fair,
0.41- 0.60	as moderate,
0.61-0.80	as substantial
0.81-1.00	as almost perfect agreement.

Figure 6: Inter-annotation Agreement for WSD

3.3. System Architecture

The general architecture of the proposed word sense disambiguation model is shown in Figure 6, and it includes four primary activities, including preprocessing for both the training and testing phases, morphological analysis, model development, and model evaluation. Because of time and resource limitations we collected 800 words and 10ksentences. Of a total of 10k sentences, 5k sentences are with ambiguous words and 5k sentences are without ambiguous words to train the proposed model. 800 words are used for sense identification during disambiguation. After collecting the data set the system accepts input sentences, including ambiguous words. Then preprocessing tasks such as tokenization, special character removal, stop word removal, and normalization is applied by using text preprocessing algorithms like AmharicSegmenter (amseg)[1] to make a suitable condition. To make the raw data received from many sources clearer information that is better suited for work, data preprocessing is crucial. In other words, it is a first step that takes all of the information that is currently available and organizes, sorts, and merges it.

For morphologically complicated languages like Amharic, where it is difficult to keep every conceivable word in WordNet, morphological analysis is crucial (Hassen, 2015). In our study with the help of morphological analysis, we were able to reduce the word's many forms into a single form or root word. To determine if a word is ambiguous or not, we first apply a morphological analyzer to the classification model. If the word is ambiguous, we once more use the model to resolve the ambiguity. With performance evaluation matrices like precision, recall, and F1 score, the model is then evaluated.

[1] https://github.com/uhh-lt/amharicprocessor

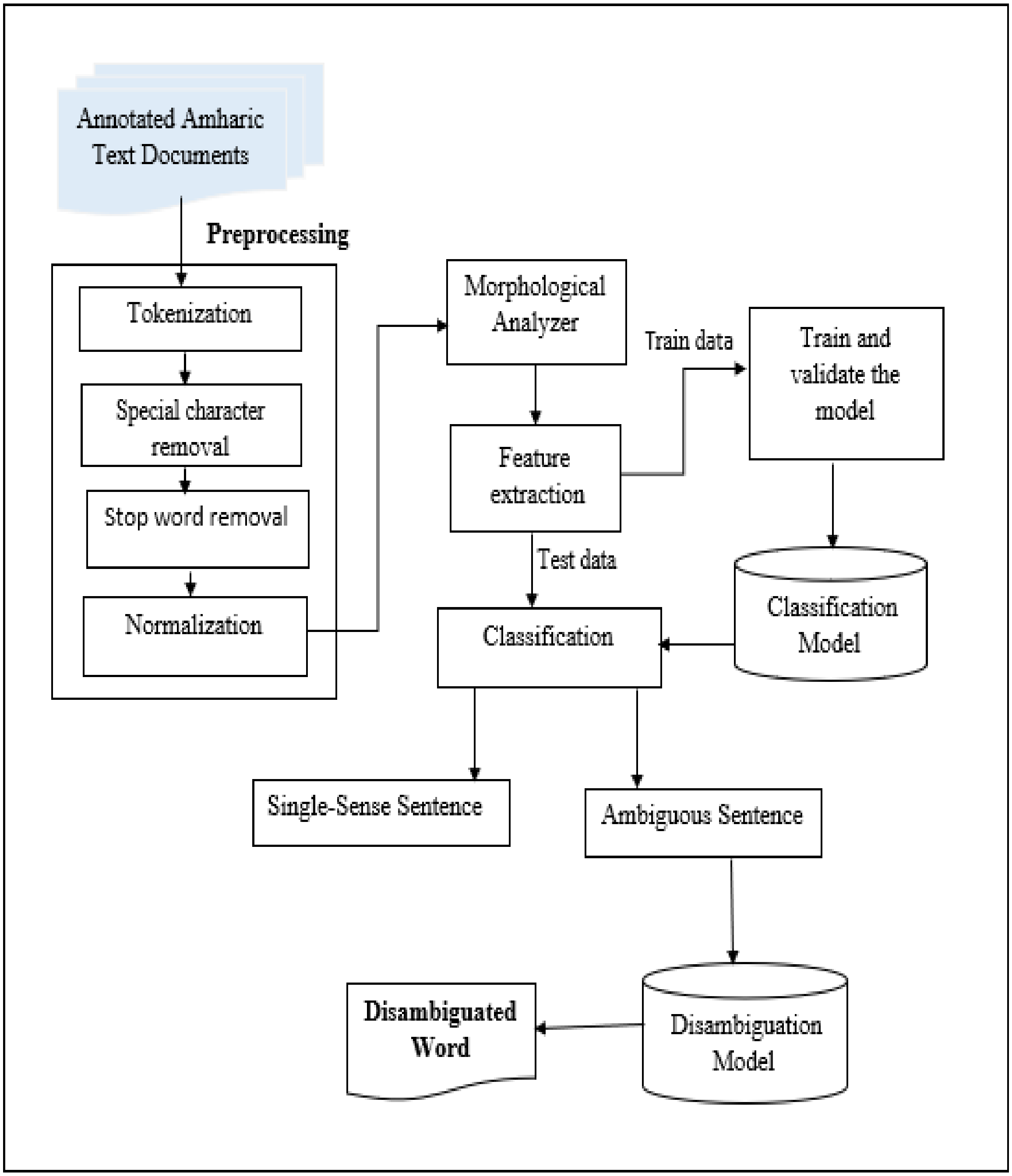


Figure 7: General flow of the proposed Amharic Word sense Disambiguation model system Architecture

3.3.1. Preprocessing of the Prepared Dataset

Information retrieval and NLP both depend on preprocessing, which is a key task (Vijayarani et al., 2020). In natural language processing preprocessing is the first step. The textual dataset used in our model is not directly analyzed by the classification techniques (Sarkar, 2019). So, the data must be preprocessed to remove noise, and easily accessed by the classification techniques. In addition, preprocessing is important to produce representative or optimal data for the word sense disambiguation model by extracting content-bearing words by involving such selected preprocessing components. Unwanted punctuation marks, stop words, numerical values, and special characters are removed in the preprocessing stage. So, tokenization, special character removal, stop words removal, and normalization is the main preprocessing steps in this study.

Tokenization

Tokenization or lexical analysis is the process of splitting the given input document into a list of words. We apply tokenization because the data written in natural language are tokenized into tokens that can be understood as separate elements. So, a document's token occurrences can be utilized directly as a vector to represent the document. Tokenization involves separating individual words from the text, omitting Amharic punctuation, including hyphens and brackets, and then returning to the list of words. In our work Amharic special characters, punctuation marks, and numbers are removed such as 'ሁለት ነጥብ'(፡)/two points/ ኮለን(,) /colon/ 'አራት ነጥብ(።)/full stop/, 'ነጠላ ሰረዝ'(፣)/semicolon/, 'ድርብ ሰረዝ' (፤)/double comma/, ጥያቄ ምልክት?'/question mark/, ቃል አጋኖ (!)/exclamation mark/,ይዘት(.)/dot/, and ትምህርተ ስላቅ(¡)/ etc. In addition, numbers like (0, 1, 2,3…9, i, ii, iii, …. x, and '፩','፫','፪','፬','፭','፮', etc.) are removed using AmharicSegmenter. Algorithm1 shows how the Amharic document changed to a sequence of words or tokens. Algorithm two also shows how special characters, numbers, and punctuation marks are removed.

```
#Tokenization
input: sentence dataset
output: non special character list of words
start
import AmharicSegmenter
load sentence dataset
    for sentence in the dataset
       tokenize sentence
       store tokenized sentence
 end
```

Algorithm1: tokenization

```
#Special character removal
input: sentence dataset
output: non special character list of words
start
import amharicSegmenter
load sentence dataset
load special characters lists
read special characters lists
split the special character in to list
     for sentence in the dataset
        for character in special characters lists
           remove character
           store non special character sentence
end
```

Algorithm 2: special character removal

Stop word removal

Stop words are common words that will likely appear in any text (Sarkar, 2019. Stop words are removed from our data set because they are less important for analysis, which does not give the meaning of the documents(Vijayarani et al., 2020). Conjunctions, articles, and prepositions are the most often used stop words in Amharic text documents and don't provide any significant information to the text. NLP applications will have a different list of stop words based on the problem they solve. We have collected stop words based on the dataset we have used and the main task that we have proposed. To increase the performance of our model and to extract ambiguous words accurately we apply the stop word removal phase. Some words in the Amharic language are considered stop words such as የ, በ, ነው, እና, ስለዚህ, ወይም, ና, ሁሉም, etc. Therefore, stop words are removed because they are not used as a keyword.

```
#stopword removal
input:  Sentence dataset
output: Non-stop word list of words
start
load Sentence dataset
load Amharic stop-word lists
read Amharic stop-word lists
    for sentence in the dataset
       split the sentence in to word list
       for word in a sentence
           if the  word is not found in Amharic stop-word list
              Remove
              store non-stop word
End
```

Algorithm 3: stop word removal

Normalization

Normalization is the process of changing a list of words into a more uniform sequence. Amharic is morphologically rich and complex; many characters are used interchangeably without affecting the meaning. For example, the character ሀ has the following similar characters (ሀ, ሃ, ሐ, ሓ, ኀ, and ኃ) that are not affecting the meaning of the given word. The word 'false' can have the following representations in Amharic (ሀሰት, ሃሰት, ሐሰት, ሓሰት, ኀሰት, ኃሰት, ሀሠት, ሃሠት, ሐሠት, ሓሠት, ኀሠት, and ኃሠት) these Amharic characters (ሀ, ሃ, ሐ, ሓ, ኀ, ኃ, ሰ, and ሠ) have the same sound but they do not make any change on the meaning of the word. They need to be converted into a single representative character "ሀ" and "ሰ" which have similar usages and different forms(Hassen, 2015). Normalization reduces such homophone variation of Amharic words to a common form. In our study, identifying and replacing Amharic alphabets that have the same usage and pronunciation but have different representations were done by using AmharicNormalizer.

```
#Normalization
input:  Sentence dataset
output: Normalized List of Words
start
import AmharicNormalizer
load Sentence dataset
read the data set
     for sentence in the dataset
        normalize sentence
        store Normalized sentence
end
```

Algorithm 4: normalization

3.3.2. Morphological Analysis

The process of breaking down words into their morphemes and assigning grammatical morphemes to grammatical categories and lexical morphemes to lexemes is known as morphological analysis (Gasser, 2009). Amharic root words can generate several lexical forms with various meanings. Affixes like prefixing, suffixing, and infixing are used in the Amharic language to produce inflectional and derivational word forms. One of the components used to reduce the different original forms of a word to a single root of this word is morphological analysis, which is utilized for determining the root morphemes of the words. It is preferable to use less memory by representing several word forms in a single format. At this stage reducing morphological variants of Amharic words by removing affixes. Which is generating the smallest unit of morphologically inflected and derivate words because many meaningful Amharic words can be generated from a single morpheme. For example, Amharic sentence "ብዙ የተማረና ስራ የሚፈልግ ሀይል ቢኖርም እንኳ መንግስት ስራ ሊፈጥርለት አልቻለም" is changed in to "ብዙ ማረ ስራ ፈለገ ሰው ሀይል ኖረ እንኳ መንግስት ስራ ፈጠረ ቻለ". Using of morphological analyzer helps the algorithm to perform better context handling because it gains a root form.

For morphologically difficult languages like Amharic, the WSD system will see large gains as a result of morphological research. because WordNet cannot possibly contain all possible worlds (Hassen, 2015). The Gasser-created Hornmorpho morphological analyzer was employed in this thesis. A group of Python scripts called HornMorpho analyze and generate words in the Amharic, Tigrinya, and Oromo languages(Gasser, 2011). Through the python interpreter, the user can interact HornMorpho with the program. HornMorpho was used to store stem words.

3.3.3. Feature Extraction

This method represents words as dense word vectors (also called word embeddings) (Sarker, 2021). The word embeddings collect more information into fewer dimensions. Note that the word embedding does not understand the text as a human would, but rather maps the statistical structure of the language used in the corpus. They aim to map semantic meaning into a geometric space. This geometric space is then called the embedding space (Sarker, 2021). This would map semantically similar words close to the embedding space like numbers.

In this phase feature extraction or word embedding technique that works better for WSD was selected. To build a deep learning model, the dataset has to be in a vector, because models do not understand texts directly but rather changed them into numeric form. In machine learning, each word in a document must be represented as a real-valued vector. Each word is converted into a vector, and each vector format is learned using deep learning neural networks After the dataset has been preprocessed tokens need to be represented in numeric value by using the feature extraction technique.

In NLP, feature extraction or word embedding is the core concept. To model and train, those embedding techniques have steps because a neural network cannot understand text directly. In the first step, it computes the similarity of words and similar words have approximately the same value vector. In the second step, make a group of similar and dissimilar words, or a word that cannot have the same characteristics as another word in a document. In the third step, feature extraction for the target group is mapped into a word vector array, such as text classification tasks. This data was then fed into the embedding layer of a model, which was used to train and predict the target group.

Word2Vec: One of the most common ways to express document vocabulary is by word embedding. It can identify a word's position in a document, its semantic and syntactic similarities, its relationship to other words, etc. Converting words or tokens into a numeric vector that can be fed into deep learning models is a key component of module development for NLP applications since deep learning accepts and interprets words in their numeric representation (Ayalew, 2021). To determine the semantic connection between the tokens in a corpus, the Word2Vec method is utilized.

Deep learning's fundamental component, however, is the automatic feature extraction that occurs without being handcrafted. To properly classify Amharic text, we convert the word into a vector using Word2vec after the preprocessing stage. We then use a vector as an input for deep learning models termed CNN and BiLSTM. We convert the word into a vector using Word2vec after the preprocessing stage.

3.4. Implementation tool and Algorithm

The proposed Amharic WSD system is implemented on the Anaconda Navigator and the Python programming language is used for development and testing. Python is an open-source scripting language that is readable, powerful, easy to learn, cross-platform, and applicable in a vast scope of NLP applications. For the experiment, we used the interactive Jupiter notebook tool.

For this research, we compared three models CNN, BiLSTM, and BERT to classify whether the word is ambiguous or not. BERT used self-attention-based transformer architecture, which, in combination with a masked language modeling target, allows to train the model to see all left and right contexts of a target word at the same time (Chawla et al., 2019). After identifying whether the word is ambiguous or not, the next task is assigning the meaning of an ambiguous word based on the context in which it is used. So, to disambiguate the ambiguous word we apply the AmRoBERTa model along with its **masked** word prediction strategy. AmRoBERTa is a recent pre-trained transfer learning approach that gives better performance in the available datasets (Yimam et al., 2021). The reason for selecting this algorithm is, that BERT-like models have an advantage over static embedding as they can accommodate different embedding representations for the same word based on its context (Yimam et al., 2021). Static embedding such as Glove and Word2Vec depends on the co-occurrence of the words in the whole corpus but in BERT, if a word is used in a different context, they will have different representations.

3.5. Evaluation

Cross-validation is a common method for evaluating model performance. It divides the training dataset into random, equal-length datasets for training and testing. This study employs evaluation metrics such as accuracy, precision, recall, and F1 score.

Accuracy: the percentage of datasets that were categorized with the correct class.

Precision: the percentage of datasets the classifier got right out of the total number of examples.

Recall: the percentage of datasets the classifier predicted for a given tag out of the total number of datasets.

F1 Score: the harmonic mean of precision and recall.

3.6. Classification Model

After preparing the dataset, and preprocessing the dataset, the classification part of the model started. Currently, transfer learning is a state-of-the-art approach in NLP and has achieved better performance in WSD. As a result, we have developed our model using a transfer learning approach.

In this study, we have used CNN (convolutional Neural Network), BI-LSTM (bi-directional long short-term memory), and BERT (Bidirectional Representation from Transformer) to detect ambiguity and the performance result is compared. We select BERT to classify whether the word is ambiguous or not. Because BERT is better than both CNN and BiLSTM algorithms for semantic understanding. This is because BERT is the first deeply bidirectional, unsupervised language representation, pre-trained using only a plain text corpus. The main limitation of the earlier research is an inability to take into account both left and right contexts of the target word, since the language model objective is generated from left to right, adding successive words to a sentence. Bidirectional LSTM, simply concatenated the left-to-right and right-to-left information, meaning that the representation couldn't take advantage of both left and right contexts simultaneously.

BERT replaces language modeling with a modified objective they called "masked language modeling". In this model, words in a sentence are randomly erased and replaced with a special token ("masked") with some small probability. Then, a Transformer is used to generate a prediction for the masked word based on the unmasked words surrounding it, both to the left and right.

BERT fine-tuning: fine-tuning the parameters is important because BERT is one of the models which contains a big neural network architecture with a large number of parameters. So, training BERT from scratch using a small dataset causes the model to have less learning or underfitting problem. We fine-tune the parameter fully or partially, for WSD we have used a total of 10m parameters. All pre-trained parameters are fine-tuned and trained these parameters for our desired Amharic WSD task.

3.7. Word sense Disambiguation model

After the sentence is classified as ambiguous by the classification model, detecting the ambiguous word from the sentence, and generating a possible sense of the target word is the main goal of this study. In our study, we used the AmRoBERTa model along with its masked word prediction strategy for the transfer learning approach. AmRoBERTa is a general purpose pre-trained model, that can be used to disambiguate the words. Even though BERT is static making but RoBERTa is trained with dynamic masking for 15% of the document then of the selected tokens, 80% are replaced by a special token called <MASk> the rest of the document stays unchanged(Liu et al., 2019). For Word sense disambiguation, we fine-tune the AmRoBERTa model with 33,297 Sentences. Which are collected from Amharic bible, Amharic Quran, and Amharic News. After fine-tuning the model, we have replaced the complex word with the token [MASK] for disambiguation purposes. Flair is one of the document embedding technique to select the most similar word for the target sentence. So, the fine-tuned AmRoBERTa model is used with flair document embedding technique, and the most similar word is selected.

3.8. Performance Metrics

The performance of the model is evaluated based on the confusion matrix: recall, precision, and F1 score for machine learning algorithms and Accuracy and loss with a graph for Deep learning algorithms.

confusion matrix: a confusion matrix is often used for binary classification tasks, showcasing how well the items in a validation set are classified and providing more details on the performance of the classifier.

True positives (TP): are positively-labeled (ambiguous sentences) samples that are correctly predicted as positive (ambiguous sentence).

False positives (FP): are negatively-labeled (single-sense sentence) samples that are incorrectly predicted as positive (ambiguous sentence).

True negatives (TN): are negatively-labeled (single-sense sentence) samples that are correctly predicted as negative (single-sense sentence).

False negatives (FN): are positively-labeled (ambiguous sentences) samples that are incorrectly predicted as negative (single-sense sentence).

Accuracy**:** accuracy is the percentage of correctly classified samples overall. To counteract the inadequacies of the accuracy measure, machine learning studies often supplement their metrics with recall, precision, and F1 score. The following definitions describe the metrics in terms of classifying the positive class.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \tag{3. 1}$$

recall: is the proportion of positively labeled samples that are successfully predicted.

$$Recall = \frac{TP}{TP + FN} \qquad 3.\ 2$$

precision: is the proportion of positively predicted samples that are labeled positive.

$$\text{Precision} = \frac{TP}{TP + FP} \qquad 3.\ 3$$

F1-score: is another accuracy metric, which is the harmonic mean of precision and recall.

$$F1 - score = \frac{2 * (precision * Recall)}{Precision + Recall} \qquad 2.\ 1$$

Where: TP is true positive, TN is a true negative, and FP is false positive

CHAPTER FOUR

EXPERIMENT AND RESULT

4.1. Overview

In this chapter, we have discussed the implementation procedure, the data set used, experimental setups, and experimental results. Deep learning models (CNN, Bi-LSTM) and transfer learning models (BERT) with selected hyperparameter values are used to build, train and test our classification model. The four evaluation criteria that were employed to assess the performance of the proposed architecture were accuracy, precision, recall, and F1 score. The result was analyzed based on these evaluation metrics.

4.2. Experimental Setup

4.2.1. Development tool

In this study, we use tools used for preprocessing, classification, model development, and model evaluation tasks called NLP Toolkit. Natural Language Processing Toolkit (NLTK) is exactly what its name implies: a Python toolkit for NLP tasks. NLTK provides access to a library for text processing operations such as tokenization, stop word removal, and normalization. Python has a lot of open-source NLP packages. It includes Jupyter Notebook, TensorFlow with deep learning library, Keras, Gensim, and other dependency necessary libraries

TensorFlow: TensorFlow is an open-source software library for numerical computation using data flow graphs, released by Google in 2015 to promote research in deep learning. Although not limited to neural networks, TensorFlow programs utilize multidimensional array data structures called tensors which serve as edges in a graph, connecting the nodes within a network (TensorFlow).

Genism: is a Python framework for vector space modeling. Genism provides APIs for using word2vec to load a model and incorporate them into our system (Models.Word2vec – Word2vec Embeddings — Genism).

Keras: it is a popular and simple deep learning framework to define, develop and evaluate the performance of the model.

4.2.2. Data set Description

At present time there is no standard sense-tagged Amharic test dataset for WSD or related research (Mulugeta, 2019). Afterward, we divided the list of these Amharic sentences into three splits, including training, validation, and test sets. For the Amharic word sense classification experiment, we have used the 80/20 rule. i.e 80% for training, 10% for validation, and 10% for testing. This technique helps in the detection of ambiguous and non-ambiguous Amharic texts.

Table 2:Word sense disambiguation dataset for training, validation, and testing

Word Sense Disambiguation sentence datasets			
Training	Validation	Testing	Total
8000	1000	1000	10,000

Table 3:Sample Ambiguous words with their sense

Word Class	word	Possible sense	Number of senses
Nouns	ቀና	1.ቀጥታ፣እውነት 2.ሰመረ፣ለማ 3.አላቀረቅርረም 4.ለመደ፣ተማረ 5.አገሩ ገበረ 6.መቅናት 7.ሰረፀ፣ገባ 8.ኮሶ አሻረ	8
Verb	ለቀመ	1.አነሳ፣ሰበሰበ 2.አረመ፣ነቀሰ 3.አጠፋ 4.ጨረሰ 5.ተማረ፣ለመደ 6.ለየ 7.በላ	7
Adverb	ገና	1.በአል 2.አልደረሰም 3.አልተፈፀመም 4.ቀድሞ 5.ዕሩር 6.ገነነ	6
	ቀድሞ	1.ፊት 2.ድሮ	2

4.2.3. Implementation and hyperparameter

In this experiment, Python programming software is used for development and testing. Because, Python is open-source scripting, which is readable, easy to learn, cross-platform and applicable language in a wide range of NLP applications. It integrated with the TensorFlow module within the anaconda navigator. Whereas Jupiter is a scientific development environment for python, it includes editing, interactive debugging, and testing. We have done our experimentation by setting environmental and hyper-parameter setups. For training and testing, we have used an NVIDIA GeForce RTX 1080/2080 Ti generations of GPU server, where each GPU has 12GB memory, with 32 CPU cores and 252 RAM to run our experiments

In our study, we use different learning hyperparameters that influenced the experimental result. Hyperparameter tuning is used to determine the right combination of hyperparameters to maximize the model performance. The Hyperparameter value that is best suited for the model is selected.

The loss function: is one of the learning hyperparameters, it predicts the model's error, which is used to measure the gradients, and the function which calculates the loss is the loss function. It is used to determine the error between the output and the target value. So, to minimize the loss in the optimization problem we used the loss function. We used **binary_crossentropy** loss functions.

Optimization: is a hyperparameter that factor into deep learning. There were several deep learning optimizers; Adam was used for our experiment. It was effective for solving our problem, having a very good learning rate rather than the others because it computes each parameter's learning evaluation.

Dropout-rate: is another hyperparameter used to reduce overfitting by dropping neurons. Dropout has seen increasing use of deep learning approaches and increased the prevention of overfitting within the training phase. To reduce overfit or much training by dropping neurons we set a dropout-rates. It randomly selects any node to be dropped within a given probability of 20 percent. We have tested our experiment on a dropout rate of 0.1, 0.2, and 0.3. Then we have selected the dropout rate of 0.2 because when the dropout rate is 0.1 model overfitting problem occurs. And if the dropout rate is 0.3 more neurons are deactivated and the underfitting problem occurred.

Batch-size: is the number of samples that will be passed through the network at one time. If the Batch-size is longer the training is quicker and the smaller Batch-size longer training. We tried to set the optimal Batch-size 64 in our research. Because if the batch-size is 32, model underfitting occurs and the batch-size is 257 there is quick learning but, the performance is decreased.

Epoch: during the training of the model, the epoch is the number of complete cycles (one forward pass and one backward pass) for data to be learned. To reduce overfitting (overlearning) and underfitting (less learning), we have used a maximum of 60 complete cycles/iterations to train the dataset for classification models (i.e CNN,BiLSTM, and BERT). After epoch 60 it is not handling new features from the dataset.

Table 4: CNN, BiLSTM, and BERT hyperparameters

No	Hyper-parameter	Size/type
1	Batch_size	64
2	Random state	120
3	Dropout	0.2
4	Activation function	relu/sigmoid
5	Epoch	40-60
6	Optimizer	Adam
7	Sequence length	50
8	Learning rate	0.00001
9	Dense layer	2
11	Embedding dimension	100x50

Table 5: Hyperparameter for RoBERTa model

No	Hyper-parameter	Size
s1	Batch_size	64
2	Num epoch	200
3	Num examples	33297
4	Number of parameters	83504416
6	Hidden layer	6
7	Attention head	12

4.3. Experimental Result

In this sub-section, we have presented the training, validation, and testing performance of CNN, BI-LSTM, and BERT as shown in (Table 4) based models for detecting ambiguous information, to identify whether the word is ambiguous or not. After identifying the ambiguous words, we apply the AmRoBERTa model for the disambiguation process. We have evaluated the three models using the same hyperparameter discussed in the experimentation setup section and the performance of each model is presented below.

4.3.1. Experimental result of CNN model

In this section, we have presented the training, validation, and testing performance of CNN by Appling the above hyperparameters. CNN model performance was measured based on the performance evaluation metrics.

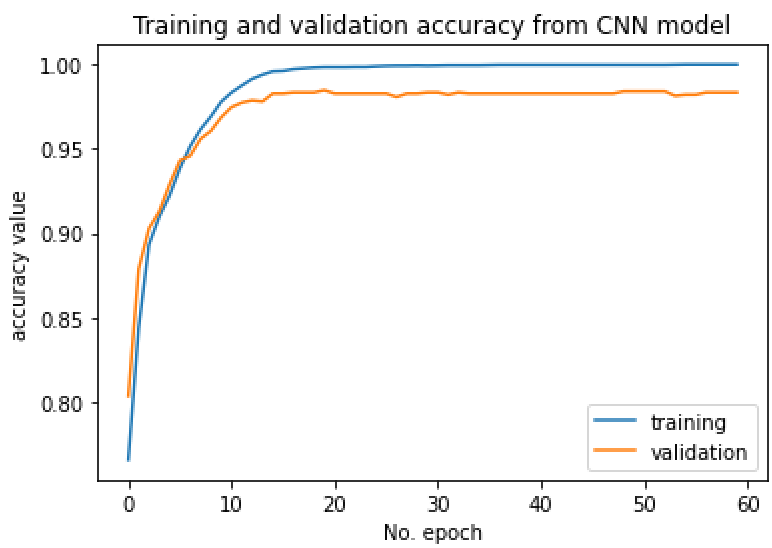


Figure 8:Training and validation Accuracy curve of CNN Model

Figure 7 shows the training accuracy and validation accuracy of CNN model Amharic word sense disambiguation problems. As shown in Figure 7; the horizontal axis represents the number of epochs or iterations, and the vertical axis represents the validation and training accuracy.

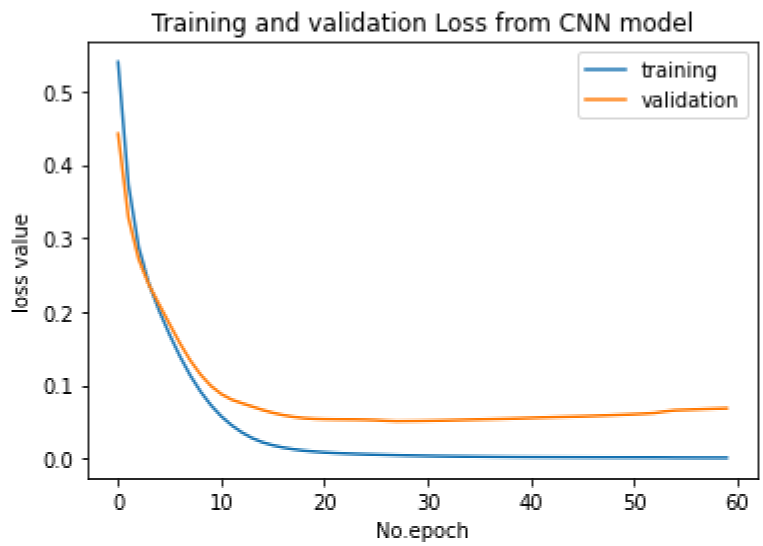


Figure 9:Tranining and validation loss curve of CNN model

On the other hand, we can see the loss value in figure 8, to know when and how the model badly predicts. The model was iterated for 60 rounds; the horizontal line(x-axis) represents the number of epochs or iterations; the vertical line(y-axis) represents the validation and training accuracy. From training and validation loss of the proposed model of a function curve, we conducted it many times.

4.3.2. Experimental result of Bi-LSEM model

Experimental results of the Bi-LSTM model were analyzed and interpreted like that of other models. We have changed the hyperparameter values that have a significant impact on our models, such as dropout value, optimizer, and learning rate values, and the values for each hyperparameter that are optimal for the model are listed in Table 1 to achieve the best model performance. We have trained the Bi-LSTM model with 2 dense layers with sigmoid activation functions and binary_crossentropy loss functions. We employed 64 neurons in the first dense, for a total of 128 neurons in both the forward and backward directions. We used, the maximum dropout rate is 0.2, the training epoch value of the model is 60, the learning rate that changes the weight of the training algorithm and we set the value of 0.00001. As shown in Table 1 we set the batch-size to 64.

Finally, as shown in figure 9, we trained to achieve optimal training and validation accuracy. One of the challenges we face when training our model is avoiding overfitting between training and validation accuracy. To solve such a problem, we use the dropout regularization technique that prevents neural networks from overfitting by modifying the cost function of the dropout. Figure 9 showed that the training and validation accuracy increased from one epoch to the next epoch. This shows that the model learns more features from one epoch than others.

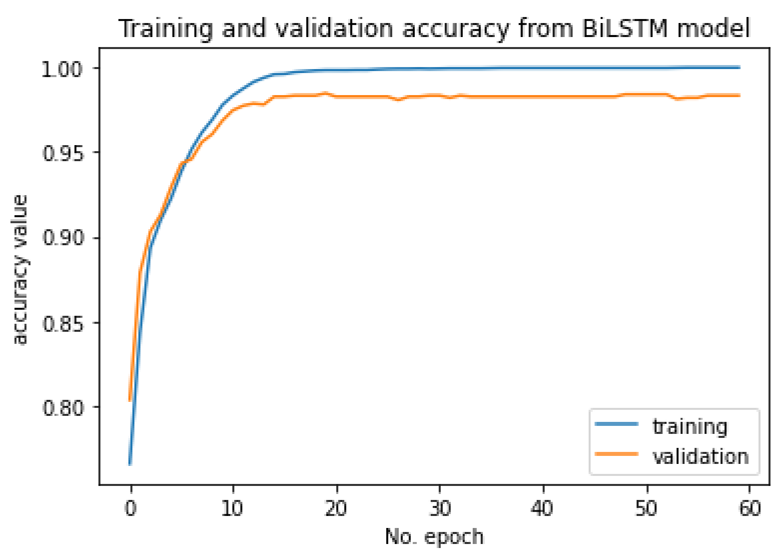


Figure 10: Training and validation Accuracy curve of Bi-LSTM Mode

The loss graph in figure 9 showed that the loss is deceased its loss when the number of epochs is increased.

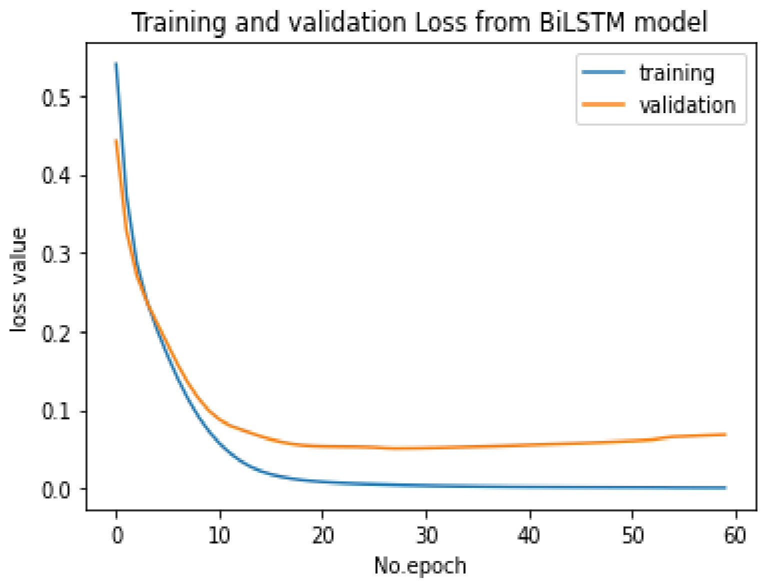


Figure 11:Tranining and validation loss curve of Bi-LSTM Model

4.3.3. Experimental result of the BERT model

Experimental results of the BERT model were analyzed and interpreted. We have used 60 epochs to train the model with a 0.00001 learning rate. To reduce overfitting or much training we set the Dropout rate to 0.2 then the model randomly selected a node and dropped with a given probability of 0.2 or 20%. We have also used the Adam optimizer. RELU for the hidden layer and Sigmond for the output layer is used as an activation function. To build the model we have used three dense layers, for the first dense we have used 64 neurons and a 0.2 dropout-rate. For the second dense 32 neurons are used. Lastly for the output layer we have used 2 neurons.

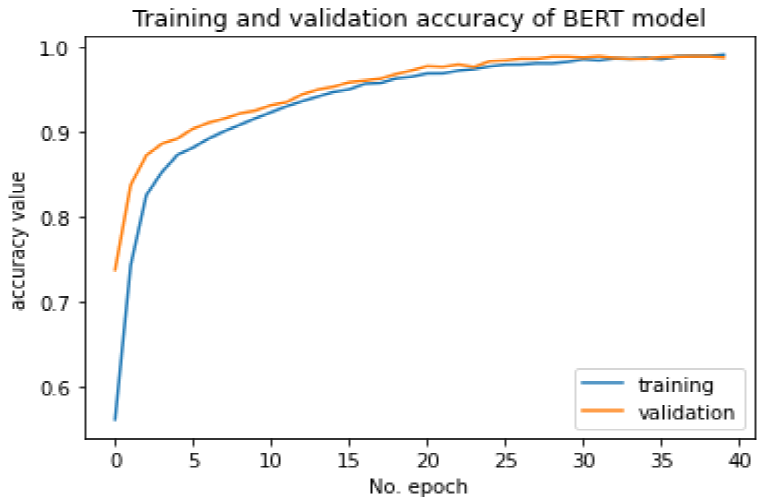


Figure 12:Tranining and validation Accuracy curve of BERT Model

As shown in figure 13, at initial, the training and validation accuracy of BERT is low. However, as the number of epochs increases the accuracy was increased too. As presented in figure 14, the training and validation loss of BERT is decreased when the number of epochs increases. Even though BERT takes much more training time than CNN and BiLSTM it has better accuracy and minimum loss.

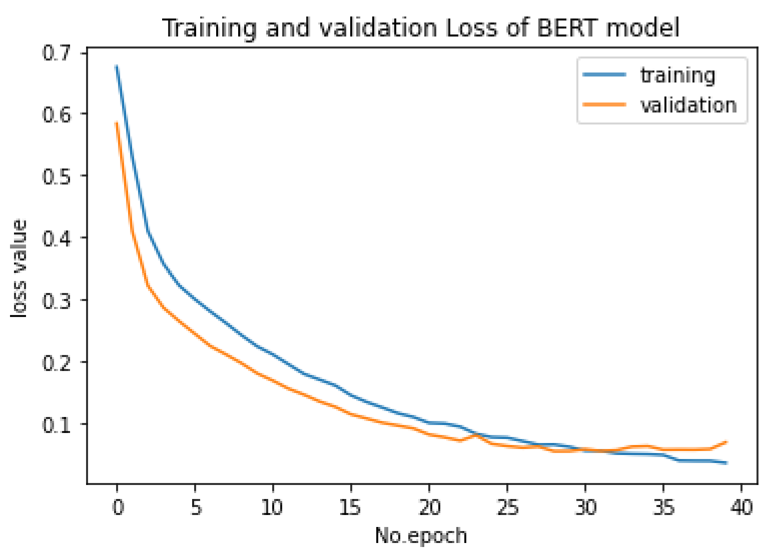


Figure 13: Training and validation loss curve of BERT Model

Table 6: *Experimental result of deep learning classification models*

Experiment	Precision (avg)	Recall (avg)	F1-score (avg)	Accuracy
CNN	0.92	0.92	0.91	0.90
BiLSTM	0.92	0.91	0.91	0.88
BERT	0.93	0.92	0.92	0.93

4.3.4. Experimental Result of Word Sense Disambiguation model

In our research, we have used the finetuned AmRoBERTa model with the FLAIR document embedding technique to disambiguate Amharic words in the given sentence.

AmRoBERTa fine-tuning: We fine-tuned the AmRoBERTa model using 33,297 sentences and 800 ambiguous words. When we train the model, we have used a maximum of eight contextual meanings for a single ambiguous word. As shown in (Table 5) our experiment is conducted using an epoch of 200 and the batch_size is 64 in NVIDIA GeForce RTX 1080/2080 Ti generations of GPU server, where each GPU has 12GB memory, with 32 CPU cores and 252 RAM to run our experiments. We have conducted our experiment with 100 and 150 epochs, and the model has an underfitting problem. Then we set it to 200 epochs which is the optimal iteration for our data set. We have also experimented with batch-size 32, 64, and 257. But we have selected batch-size 64 as the optimal batch size because when the batch-size is below 64 it takes more training time. When the batch-size is more than 64, there is faster training but the performance is low.

Word Sense Disambiguation: For word sense disambiguation, we have used the finetuned pre-trained contextual model to disambiguate the correct sense of the ambiguous words. We have used the fine-tuned AmRoBERTa model with the FLAIR sentence embedding technique. For the disambiguation task, we have followed a similar approach as Huang et al. (2019), where we have to prepare the target sentence and gloss sentence pairs. However, there is no WordNet for Amharic to employ for this task. Hence, we have selected 10 words that are previously annotated using the WebAnno annotation tool. These words are ዋና (Wana), መንገድ (Menged), ሳለ (Sale), አካል (Akal), ዋጋ (Waga), ገና (Gena), ቀና (Qena), ሀቅ (haq), ሃይል (Hayil), and ልክ (Lik). The different senses of these words along the English translation are provided in Table 3 and more details in Appendix D . Then we constructed a gloss for 10 words, which contains the ambiguous word and possible senses with examples sentences. For each word, we construct at most eight senses with example sentences. During disambiguation, we select a target sentence that contains ambiguous words and the sense is already annotated by the annotators. Then we use the FLAIR sentence embedding with the finetuned contextual pre-trained model to compute the similarity between the target sentence and the glosses. The sense which has a high similarity value with the target sentence would be the correct meaning of the ambiguous word. So based on the given sentence in the gloss, the model disambiguates the target word into its correct sense. Let's see these three sentences as examples.

Example 1: The sentence "እያንዳንዱ ዋና ሃሳብ አንድ አንቀፅ ውስጥ ሰፋሯል።" is disambiguate as follow.

```
Target sentence: እያንዳንዱ ዋና ሐሳብ ራሱን በቻለ አንድ አንቀጽ ውስጥ ሰፍሯል ።

ጭብጥ: የሰብሰባውን ዋና ሃሳብ አስረዳ ። ዋናውን ነገር ብቻ ንገረኝ 0.5702
ዐይነተኛ:  ተወካይ  ሆኖም የተለየ ልብስ ለብሳ ዋና አጋፋሪ ሆና ድግሱን ስታሞቀው ውላለች ። የቡና ዋና ጥቅም ማነቃቃት ነው ። አቋራጩን መንገድ ትቶ በዋናው መንገድ ማጣ ። 0.5347
መሪ ፤ ሀላፊ ፤ :  አለቃ የሆቴሉ ዋና ሀላፊ ጥራልኝ ። ዋና የሌለው ጦር ቶሎ ይሸነፋል ።  0.3580
የባሕር ላይ ስፖርት: ይሁን እንጂ ውሃ ዋና በአንድ ዘመን ብልጭ ብሎ የጠፋ ስፖርት ሆኗል ። ዋና መዋኘት ስለቻለ ከባህሩ ወጣ ። ከብት የቋም ዋና ስለሚዋኝ ውሀ አይበላውም  እያሉ የኛ አገር ሰ
ዎች ሲጫወቱ ብዙ ጊዜ ሰምቻለሁ ። 0.3449
```

Based on the result of our experiment, for the target sentence" እያንዳንዱ **ዋና** ሐሳብ ራሱን በቻለ አንድ አንቀፅ ውስጥ ሰፍሯል ።" the correct sense of the ambiguous word **ዋና** is **ጭብጥ**, as it has higher similarity with the target sentence (0.5702) compared to the other senses, which are **አይነተኛ** and ***መሪ/ሀላፊ*** with similarity scores of 0.5347 and 0.3580 respectively.

Example 2: The sentence "ረብሻው ከተረጋጋ በኋላ የተወሰኑ ታክሲዎች መንገድ ላይ መታየት በመጀመራቸው ህዝቡ ተደሰተ" is also disambiguated as follow.

```
Target sentence: ረብሻው ከተረጋጋ በኋላ የተወሰኑ ታክሲዎች መንገድ ላይ መታየት በመጀመራቸው ህዝቡ ተደሰተ

ጎዳና: በአንድ ወቅት መንገድ ዳር ለተሰበሰቡ ሰዎች ፊልሙን ማሳየት ጀመርን ። ከዚህ ውሳኔ በኋላ የተወሰኑ ታክሲዎች መንገድ ላይ መታየት ጀምረዋል ። መንገዱ ስለሚያስፈራ ሸኘኝ ። 0.8098
አካሄድ: ትክክለኛና ፍትሃዊ ጥያቄ ግን በሰላማዊ መንገድ መቅረብ ነበረበት ። ሰው በሁለት መንገድ ከጥፋቱ ይማራል ። በትክክለኛው መንገድ መነገድ አለብህ ። እስኪ በትክክለኛው መንገድ ቁትሩን ደምረ
ው ። 0.4688
አስተሳሰብ: ከሆነ ይህንን መጽሔት ከሰጡህ ሰዎች ጋር የምትመሳሰልበት መንገድ አለ ። በምን መንገድ ልታስረዳኝ ትችላለህ ። 0.4052
ብልሀት ፤ ዘዴ: ይህን ለማድረግ የሚያስችለው አንደኛው መንገድ የስብከቱ ሥራችን ነው ። አራተኛው የውጭ ምንዛሪ ማግኛ መንገድ የመንግሥት ኃዋላ የሚባለው ነው ። ኢየሱስ ክርስቶስ እውነተኛ ነፃ
ነት ማግኘት የሚቻልበትን ቀላል መንገድ ጠቁሞናል ። 0.3305
አሰራር: በሰላማዊ መንገድ ነፃና ዴሞክራሲያዊ ምርጫ መኖር አለበት ። 0.3136
ሁኔታ: በዚህ መንገድ የአምላክን ሕግ መጣሳቸው ኃጢአተኞች እንዲሆኑ አደረጋቸው ። የዘካርያስ የመናገር ችሎታ በተአምራዊ መንገድ ተመለሰለት ። 0.2925
```

Based on the result of our experiment, for the target sentence "ረብሻው ከተረጋጋ በኋላ የተወሰኑ ታክሲዎች መንገድ ላይ መታየት በመጀመራቸው ህዝቡ ተደሰተ።" the correct sense of the ambiguous word ***መንገድ*** is ***ጎዳና***, as it has higher similarity with the target sentence (0.8098) compared to the other senses, which are **አካሄድ, አስተሳሰብ, አሰራር, and ሁኔታ** with similarity scores of 0.4688, 0.4052, 0.3305,0.3136 and 0.2925 respectively.

Example 3: The sentence "ልጅ ዶክተር አብይን ስለሚወደው ሳለው።" is disambiguated as follow.

```
Target sentence: ልጅ ዶክተር አብይን ስለሚወደው ሳለው

ወረቀት ላይ አሰፈረው: ወንድሜ የጃንሆይን ሥዕል ሳለ ። የሳለውን ሰዕል ለጨረታ አቀረበው ። 0.5582
ኡኹ ! ኡኹ ! አለ:  ይህ ልጅ ጉንፋን ይዞታል መሰለኝ  ይስላልና ። ከኮሮና ቢያገግምም በጣም እየሳለ ነው ።  0.4615
በሞረድ ወይም በለሆቴ ስለትን አወጣ: ቢላዎን ሳለ  እንጨት ከመፍለጡ በፊት መጥረቢአውን ሳለ ። 0.4134
እያለ: እንደነበረ ዘካርያስ በቤተ መቅደሱ ውስጥ እያገለገለ ሳለ ገብርኤል በዕጣን መሠዊያው አጠገብ በድንገት ተገለጠ ። የምጣዱ ሳለ የእንቅቡ ተንጣጣ ። ጥሩ ሆኖ ሳለ ዋጋው ተወደደ ። 0.4074
```

Based on the result of our experiment, for the target sentence "ልጅ ዶክተር አብይን ስለሚወደው ሳለው።" the correct sense of the ambiguous word **ሳለ** is ***ወረቀት ላይ አሰፈረ*** as it has higher similarity with the target sentence (0.5582) compared to the other senses, which are ***ኡህ ኡህ አለ, ስለት አወጣ,*** and **እያለ** with similarity scores of 0.4615, 0.4134, and 0.4074 respectively.

4.3.5. Discussion

In this work, we propose Amharic word sense disambiguation model by experimenting with deep learning and transfer learning algorithms. As there is no standard sense-tagged Amharic text dataset for the Amharic WSD task, we first compiled 800 ambiguous words from different sources, including the Amharic dictionary, Amharic textbooks (Grade 7-12), and the Abissinica online dictionary. Furthermore, we collect more than 33k sentences that contain those ambiguous words. The 33k sentences are used to finetune our transformer-based RoBERTa model (AmRoBERTa).

We conduct two types of annotation for our WSD experiments. First, to know if the collected dataset contains the 7 types of word relations (synonymy, hyponymy, hypernymy, meronomy, holonomy, toponymy, and homonymy), we annotate 10k sentences using linguistic experts. Using these datasets, we have conducted classification experiments for word relation using the CNN, BiLSTM, and BERT models.

We have set a few parameters during model configuration that are suitable for managing the feature of our datasets, such as a dense layer, dropout layer, number of neurons in each dense, learning rate, and activation function. For each of the three models, we run three experiments based on the selected hyperparameters, such as CNN, BiLSTM, and BERT. These three models score classification performance at 90%, 88%, and 93% respectively. Based on our experiment BERT model has better performance, because we have fine-tuned the layers the of BERT model. This helps to be easily trainable with a limited dataset for our task. BERT has also used a self-attention mechanism and the hidden layer process the input sequence once.

Lastly, For the Word sense disambiguation (WSD), we first finetuned the AmRoBERTa model. To finetune the AmRoBERTa model, we have used an NVIDIA GeForce RTX 1080/2080 Ti generation of GPU server, where each GPU has 12GB memory, with 32 CPU cores and 252 RAM to run our experiments. We fine-tuned AmRoBERTa models using 800 ambiguous words and 33,297 sentences with a window size of 6. For each ambiguous word, a maximum of 8 senses is used. For the WSD disambiguation experiment, we first choose 10 target words and annotate a total of 1000 sentences with their correct sense using the WebAnno annotation tool. Each sentence with one target ambiguous word is annotated by two users and one curator (adjudicator). As preparing glosses for each sense is time taking, we prepare 100 glosses for the selected 10 targets. These words are ዋና፣ መንገድ፣ ሳለ፣ አካል፣ ዋጋ፣ ገና፣ ቀና፣ ሀቅ፣ ሃይል and ልክ.

For the WSD task, we have employed the FLAIR document embedding framework to embed the target sentences and glosses separately. We then compute the similarity of the target sentence with the glosses embedding. The gloss with the higher score disambiguates the target sentence. Our model was able to achieve an F1 score of 71%. Due to time constraints and the lack of Amharic WordNet, we could not experiment with a large number of training datasets. In the future, we plan to at least compile glosses for the 1000 sentences annotated using WebAnno and report the performance

Error Analysis: In order to do error analysis on the models, we also employed a confusion matrix to determine the proportion of testing samples that are correctly classified according to class and wrongly classified. Machine learning is becoming an important approach for analyzing massive amounts of data and identifying specific trends and patterns. In some circumstances, these models have the potential for bias and incorrect prediction. As a result, we described model error analysis for the Amharic word sense disambiguation model that we trained. As we can see from the test data prediction results, the models have some missed predictions. The presence of test data terms in both ambiguous and non-ambiguous labeled training datasets contributes to the model's incorrect predictions. When we see the sentence “የሞያተኛው ዋጋ እስከነገ ድረስ አይደርብህ” Its actual label was ambiguous. However, all three deep learning models predict it as non-ambiguous due to the existence of the words “ሞያ፣ ነገ፣ደረሰ፣ and አደረ” in non- ambiguous training dataset more frequent than ambiguous dataset. During the error analysis phase, we were able to see that there was model overfitting. As we can deduct from our model network, overfitting results from the dropout-rate. To control and fix this problem we tried to add several dropout-rate values during model building. By applying this technique, the overfitting problem is handled. The underfitting problem is also fixed by controlling the number of epochs. Especially, transfer learning approach models need many numbers of epochs to train the model, otherwise underfitting occurs.

CHAPTER FIVE

CONCLUSION

5.1. Conclusion

This study has developed an Amharic word sense disambiguation model by using a transfer learning approach. The process of identifying the correct meaning based on its context is known as word sense disambiguation. WSD is improving the performance of different NLP applications like machine translation so, to advance NLP research WSD is important. In addition, WSD will be abasis to build Amharic WordNet. These issues motivated us to conduct this research.

At present time there is no standard sense-tagged Amharic text dataset for Amharic WSD or related research. So, we have collected 10k sentences from Amharic news, Amharic dictionary, Amharic Quran, Amharic bible, and Amharic textbooks. For the Amharic WSD task, we have collected 800 ambiguous words from different sources such as Amharic dictionaries. A total of 33,297 sentences are used to finetune the AmRoBERTa model (transfer learning).

In our study, we have compared different models to select the most suitable model for WSD classification. To select the best fit model, we have conducted different experiments. For the word relation classification task, we have experimented with CNN, BiLSTM, and BERT algorithms with 2 dense layers and a sigmoid activation function. According to the results, CNN, Bi-LSTM, and BERT obtained 90 %, 88 %, and 93 % accuracy respectively. Based on our findings, the model based on BERT has achieved the vesting result.

As AmRoBERTa is a general-purpose pre-trained language model, we have fine-tuned it with 33,294 sentences and 800 ambiguous words. We have employed the FLAIR document embedding framework to embed the target sentences and glosses separately. We then compute the similarity of the target sentence with the glosses embedding. The gloss with the higher score disambiguates the target sentence. Our model was able to achieve an F1 score of 71%. Due to time constraints and the lack of Amharic WordNet, we could not experiment with a large number of training datasets. In the future, we plan to at least compile glosses for the 1000 sentences annotated using WebAnno and report the performance.

5.2. Contribution

The main contributions of this study include:

- We have developed three deep learning models CNN, Bi-LSTM, and a pre-trained deep learning language model called BERT; those are useful for the classification of Amharic word sense ambiguity problems.
- In addition, for the disambiguation task, the AmRoBERTa model is also employed.
- Recently, Amharic word sense disambiguation was done on 159 ambiguous words, and 2161 sentences. However, in this research, we have prepared a dataset of size 10k Amharic sentences and a total of 800 ambiguous words which can be used as a benchmark for further future research work on the area.
- Word sense disambiguation is the main component for different NLP applications such as machine translation, and Information retrieval. Our model can be directly integrated into downstream NLP applications.

5.3. Recommendation

The recommendation of this study includes the following points:

- In our study, because of time and resource limitations we did not consider syntactic, phonological, and referential ambiguity, so for the future we recommended to consider these ambiguity types.
- Other language WSD researchers in high-linguistic language use a linguistic resource like WordNet, and Gloss but for Amharic language, these linguistic resources are not developed, we recommended including this resource in future work.
- Considering grammar and spelling checker is also our recommendation.

REFERENCES

Aggarwal, C. C. (2014). Transfer Learning. In *Igi Global* (Pages, 657–665). Https://Doi.Org/10.1201/B17320

Agirre, E., & Edmonds, P. (2006). Word Sense Disambiguation: Algorithms And Applications. In *Text Speech And Language Technology* (Vol. 33, Number. 33).

Alian, M., Awajan, A., & Al-Kouz, A. (2016). Arabic Word Sense Disambiguation Using Wikipedia. *Researchgate*, Volume, *12*, *No 1*, Pages, 61–66. Https://Doi.Org/10.21700/Ijcis.2016.108

Amare, G. (2001). Towards The Analysis Of Ambiguity In Amharic. *Published By : Institute Of Ethiopian. Available On Jstor For This Article : Studies Stable Url : Https://Www.Jstor.Org*, *34*(2), 35–56.

Assemu, S. (2011). Unsupervised Machine Learning Approach For Word Sense Disambiguation To Amharic Words. *Addis Ababa, Ethiopia. Masters Thesis Addis Ababa University, Ethiopia*,.

Ayalew, A. (2021). Design Event Extraction Model From Amharic Texts Using Deep Learning Approach. *Bahir Dar University, Masters Thesis Bahir Dar Ethiopia*.

Bakx, G. E. (2006). Machine Learning Techniques For Word Sense Disambiguation. *Universital Politecnica De Cataunya,Barcelona*, Pages, 1–172. Https://Www.Lsi.Upc.Edu/~Escudero/Wsd/06-Tesi.Pdf

Bouhriz, N., Habib, E., & Lahmar, B. (2016). Word Sense Disambiguation Approach For Arabic Text. *(Ijacsa) International Journal Of Advanced Computer Science And Applications*, Volume, *7 Number*, 4, Pages, 381–385.

Braz, F. A., Da Silva, N. C., De Campos, T. E., Chaves, F. B. S., Ferreira, M. H. S., Inazawa, P. H., Coelho, V. H. D., Sukiennik, B. P., De Almeida, A. P. G. S., Vidal, F. B., Bezerra, D. A., Gusmao, D. B., Ziegler, G. G., Fernandes, R. V. C., Zumblick, R., & Peixoto, F. H. (2018). Document Classification Using A Bi-Lstm To Unclog Brazil's Supreme Court. *Researchgate*, Http://Arxiv.Org/Abs/1811.11569

Chawla, A., Biemann, C., Wiedemann, G., & Remus, S. (2019). Does Bert Make Any Sense? Interpretable Word Sense Disambiguation With Contextualized Embeddings. *Arxive*.

Desmond Bala Bisandu. (2019). Design Science Research Methodology In Computer Science And Information Systems. *Researchgate*. Https://Www.Researchgate.Net/Publication/330041672_Design_Science_Research_Methodology_In_Computer_Science_And_Information_Systems

Devlin, K. (2018). *Bert: Pre-Training Of Deep Bidirectional Transformers For Language Understanding*. *MLM*.

E.Agirre. (2006). Word Sense Disambiguation Algorithm And Application. In *Springer* (Vol. 33).

El-Razzaz, M., Fakhr, M. W., & Maghraby, F. A. (2021). Arabic Gloss Wsd Using Bert. *Applied Sciences(Mdpi)*.

Ezen-Can, A. (2020). A Comparison Of Lstm And Bert For Small Corpus. *Arxive*, *Sep*, 1–12. Http://Arxiv.Org/Abs/2009.05451

Gasser, M. (2009). Semitic Morphological Analysis And Generation Using Finite State Transducers With Feature Structures. *Eacl 2009 - 12th Conference Of The European Chapter Of The Association For Computational Linguistics, Proceedings*, *April*, 309–317. Https://Doi.Org/10.3115/1609067.1609101

Gasser, M. (2011). Hornmorpho: A System For Morphological Processing Of Amharic, Oromo, And Tigrinya. *Conference On Human Language Technology For Development, April 2011*, 94–99.

Getaneh, M. (2020). Amharic Wordnet Construction Using Word Embedding. *Masters Thesis Addis Ababa University, Addis Ababa, Ethiopia*, *46*(182), 291–312. Https://Doi.Org/10.4000/Etudesafricaines.5928

Gezmu, A. M., Nürnberger, A., & Seyoum, B. E. (2019). Portable Spelling Corrector For A Less-Resourced Language: Amharic. *Lrec 2018 - 11th International Conference On Language Resources And Evaluation*, 4127–4132.

Ghobadipasha, N. (2019). Translation Versus Language Model Pre-Training Objectives For Word Sense Disambiguation. *Masters Thesis Simon Fraser University, Thehran*.

Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., & De, D. (2019). Fundamental Concepts Of Convolutional Neural Network. In *Intelligent Systems Reference Library* (Vol. 172, Issue January). Https://Doi.Org/10.1007/978-3-030-32644-9_36

Hadiwinoto, C., Ng, H. T., & Gan, W. C. (2018). *Improved Word Sense Disambiguation Using Pre-Trained Contextualized Word Representations*.

Hassen, S. (2015). Amharic Word Sense Disambiguation Using Wordnet. *Addis Ababa, Ethiopia. Masters Thesis Addis Ababa University, Ethiopia*.

Huang, L., Sun, C., & Qiu, X. (2019). Glossbert : Bert For Word Sense Disambiguation With Gloss Knowledge. *3509–3514.*, 3509–3514.

Ide, N., & Vronis, J. (1998). Introduction To The Special Issue On Word Sense Disambiguation : The State Of The Art. *Association For Computational Linguistics, Poughekeepsie, New York*, *24, Number*, 12606–0520.

Kassie, T. (2009). Word Sense Disambiguation For Amharic Text Retrieval: A Case Study Legal Documents. *Addis Ababa, Ethiopia. Masters Thesis Addis Ababa University, Ethiopia,*.

Kleenankandy, J., & K A, A. N. (2020). An Enhanced Tree-Lstm Architecture For Sentence Semantic Modeling Using Typed Dependencies. *Information Processing And Management*, *57*(6). Https://Doi.Org/10.1016/J.Ipm.2020.102362

Lewes, G. H. (2015). Support Vector Machines For Classification. *Analytics Vidharia*, *January*. Https://Doi.Org/10.1007/978-1-4302-5990-9

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *Roberta: A Robustly Optimized Bert Pretraining Approach*. *1*. Http://Arxiv.Org/Abs/1907.11692

Majumder, P. (2021). Word Sense Disambiguation: Importance In Natural Language Processing. *Analytics Vidhya*, pages,114–120.

Mathew, A., Amudha, P., & Sivakumari, S. (2021). Deep Learning Techniques: An Overview. *Advances In Intelligent Systems And Computing*, *1141*(August 2020), 599–608. Https://Doi.Org/10.1007/978-981-15-3383-9_54

Meckelburg, A. (2018). From "Subject To Citizen"? History, Identity And Minority Citizenship: The Case Of The Mao And Komo Of Western Ethiopia. *Aethiopica*, *20*. Https://Doi.Org/10.15460/Aethiopica.20.1.10

Mekonnen, S. (2010). Word Sense Disambiguation For Amharic Text: A Machine Learning Approach. *Addis Ababa, Ethiopia. Masters Thesis Addis Ababa University, Ethiopia,*.

Meshesha, M., & Jawahar, C. (2008). Indigenous Scripts Of African Languages. *Indilinga: African Journal Of Indigenous Knowledge Systems*, *6*(2). Https://Doi.Org/10.4314/Indilinga.V6i2.26422

Miangah, T. M., & Khalafi, A. D. (2005). Word Sense Disambiguation Using Target Language Corpus In A Machine Translation System. *Literary And Linguistics Computing*, *20*(2), 237–249. Https://Doi.Org/10.1093/Llc/Fqi029

Mindaye, T., Sahlemariam, M., & Kassie, T. (2010). The Need For Amharic Wordnet. *Global Wordnet Conference, Gwc 2010*.

Mosavi, A., & Ardabili, S. (2020). *List Of Deep Learning Models*. *2*, 202–214.

Mosavi, A., Ardabili, S., & Varkonyi-Koczy, A. R. (2019). List Of Deep Learning Layers. *Preprints*, *August*, 1–28. Https://Doi.Org/10.20944/Preprints201908.0152.V1

Mubatada'i, N. R., Barakbah, A. R., & Helen, A. (2019). Semantic Information Retrieval For Scientific Experimental Papers With Knowlege Based Feature Extraction. *Inovtek Polbeng - Seri Informatika*, *4*(1), 76. Https://Doi.Org/10.35314/Isi.V4i1.885

Mulugeta, M. (2019). Word Sense Disambiguation For Amharic Sentences Using Wordnet Hierarchy. *Masters Thesis, Bhir Dar University, Bahir Dar, Ethiopia*, *July*, 10–80.

Navigli, R. (2009). Word Sense Disambiguation: A Survey. *Acm Computing Surveys*, *41*(2). Https://Doi.Org/10.1145/1459352.1459355

Pal, A. R., & Saha, D. (2015). Word Sense Disambiguation: A Survey. *Researchgate*, *5*(3), Page, 1-16.

Ranjan Pal, A., & Saha, D. (2015). Word Sense Disambiguation: A Survey. *International Journal Of Control Theory And Computer Modeling*, *5*(3), 1–16.

Https://Doi.Org/10.5121/Ijctcm.2015.5301

Reta, B. (2015). Application Of Parts-Of-Speech Tagged Corpus To Improve The Performance Of Word Sense Disambiguation: The Case Of Amharic. *Masters Thesis Addis Ababa University, Addis Ababa, Ethiopia*.S

Ruder, S., Peters, M., Swayamdipta, S., & Wolf, T. (2019). Transfer Learning In Nlp. *Proceedings Of The 2019 Conference Of The North American Chapter Of The Association For Computational Linguistics*, *2010*, 15–18. Https://Aclanthology.Org/N19-5004/

Saeed, A., Nawab, R. M. A., Stevenson, M., & Rayson, P. (2019). A Word Sense Disambiguation Corpus For Urdu. *Language Resources And Evaluation*, *53*(3), 397–418. Https://Doi.Org/10.1007/S10579-018-9438-7

Salawu, A., & Aseres, A. (2015). Language Policy, Ideologies, Power, And The Ethiopian Media. *University Of South Africa(Unisa)*, *41*(1), 71–89. Https://Doi.Org/10.1080/02500167.2015.1018288

Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview On Techniques, Taxonomy, Applications, And Research Directions. *Sn Computer Science*, *2*(6), 1–20. Https://Doi.Org/10.1007/S42979-021-00815-1

Senay, D. (2021). Automatic Amharic Word Sense Disambiguation Model At Sentence Level By Deep Learning Approach. *Masters Thesis Bahir Dar University, Bahir Dar, Ethiopia*, *July* 2013–2015.

Sonja, M. R.-. (2018). High-Quality Elmo Embeddings For Seven Less-Resourced Languages. *Arxive*.

Siraj, D. (2017). A Generic Approach Towards All Words Amharic Word Sense

Disambiguation. Masters Thesis Adis Ababa University, Adis Ababa, Ethiopia. *Masters Thesis Addis Ababa University, Addis Ababa, Ethiopia*.

Tadesse. (2021). Word Sense Disambiguation For Wolaita Language Using Machine Learning Approach. *Masters Thesis Adama University, Adama Ethiopia.*

Taneja, S., Gupta, C., Goyal, K., & Gureja, D. (2014). An Enhanced K-Nearest Neighbor Algorithm Using Information Gain And Clustering. *International Conference On Advanced Computing And Communication Technologies, Acct*, *February 2016*, 325–329. Https://Doi.Org/10.1109/Acct.2014.22

Tesema, W., Tesfaye, D., & Kibebew, T. (2016). Towards The Sense Disambiguation Of Afan Oromo Words Using Hybrid Approach (Unsupervised Machine Learning And Rule Based). *Ethiopian Journal Of Education And Sciences*, *12*(1), 61-77–77.

Tsou, A. Y., Treadwell, J. R., Erinoff, E., & Schoelles, K. (2020). Machine Learning For Screening Prioritization In Systematic Reviews: Comparative Performance Of Abstrackr And Eppi-Reviewer. *Systematic Reviews*, *9*(1), 1–14. Https://Doi.Org/10.1186/S13643-020-01324-7

Vijayarani, S., Ilamathi, M. J., & Nithya, M. (2020). *Preprocessing Techniques For Text Mining - An Overview Preprocessing Techniques For Text Mining - An Overview*. *February*.

Wassie, G., Ramesh, B., Teferra, S., & Meshesha, M. (2014). A Word Sense Disambiguation Model For Amharic Words Using Semi-Supervised Learning Paradigm. *Researchgate*,Volume *3*,*Number* 3, 147. Https://Doi.Org/10.4314/Star.V3i3.25

Yap, B. P., Koh, A., & Chng, E. S. (2020). *Adapting Bert For Word Sense Disambiguation With Gloss Selection Objective And Example Sentences*. Findings Ofthe Association For Computational Linguistics, Pages, 41–46.

Yimam, S. M., Ayele, A. A., Venkatesh, G., Gashaw, I., & Biemann, C. (2021). Introducing Various Semantic Models For Amharic: Experimentation And Evaluation With Multiple Tasks And Datasets. *Future Internet*, Volume *13 Number* 11. Https://Doi.Org/10.3390/Fi13110275

Zhou, X., & Han, H. (2005). Survey Of Word Sense Disambiguation Approaches. *Proceedings Of The Eighteenth International Florida Artificial Intelligence Research Society Conference, Flairs 2005 - Recent Advances In Artificial Intelligence*, 307–312.

ሃረገወይን፤ፈትለወርቅከበደ፣ጽጌ. (1993). *የኢትዮጵያ ቋንቋዎች ጥናትና ምርምር ማዕከል % አማርኛ መዝገበ ቃላት፡፡ አዲስ አበባ ፤ አርክቲክ ማተሚያ ቤት. አዲስ አበባ ዩኒቨርሲቲ,Ethiopia.*

APPENDICES

Appendix A: Dataset Reviewer Letter

To whom it concerns

We write this letter to inform that we have reviewed a research work dataset of a student who studied at Bahir Dar Institute of Technology (BiT), Faculty of Computing, department of Information Technology MSc student. '**Nelma Mossa**' was requested us to review her research dataset and the annotation guidelines she prepared for her research work on the title "**Amharic sentence-level Word Sense Disambiguation using Transfer Learning**". We have reviewed the annotation guideline prepared and we reviewed the dataset, which is pre-annotated using the guideline prepared.

Name of the reviewer: [illegible]

Signature:

Stamp:

Email: emubirhanu16@gmail.com

Phone: 0934189009

Appendix B: Dataset Annotation Guideline

[illegible]

- [illegible] (Amharic Ambiguous words) [illegible] = (This is an evaluation of the validity of Amharic ambiguous words for Ethiopian language and literature experts).

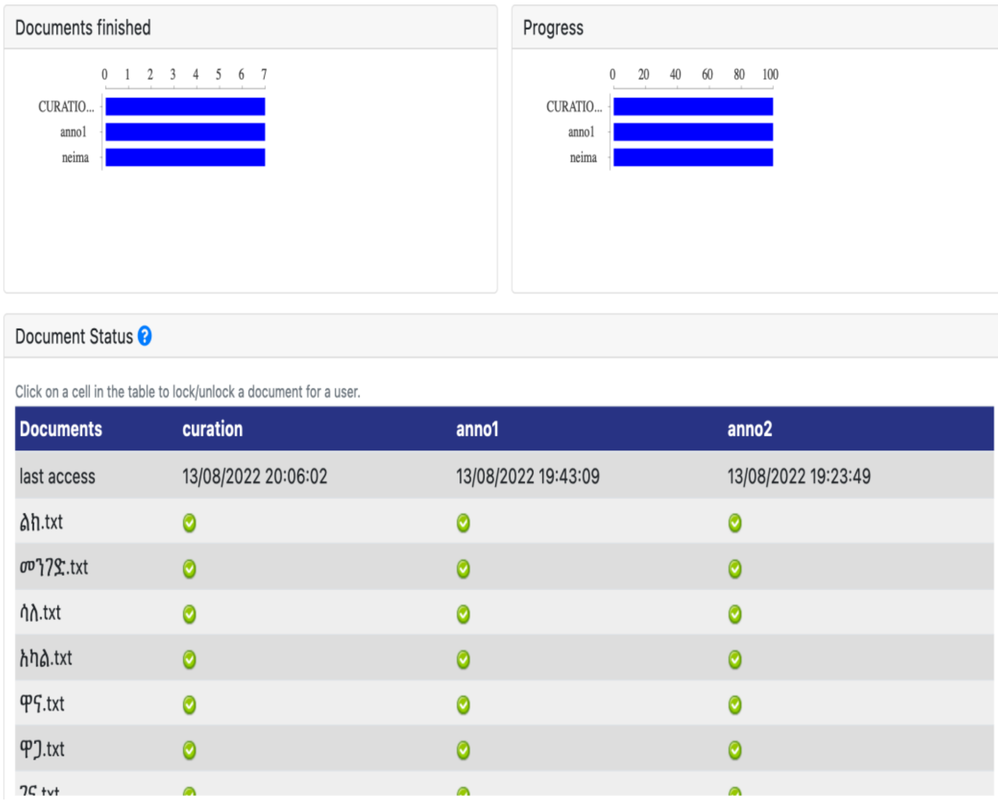
[illegible]

1. [illegible] (Read and understand sentences correctly)
2. [illegible] (Understand the stated statement from the description and mark '√' only in relation to the relevant relationship) "[illegible]!" (please notice that each of your answers has a great effect on our study).

0. **Synonymy:** Amharic words, such as "[illegible]", "[illegible]" and "[illegible]" are synonymous meaning all these words are the same function and meaning.
1. **Homonym:** [illegible] homonym [illegible] Homonym [illegible] =
2. **Meronom/Holonom:** [illegible] Meronomy (part–whole relationship) [illegible] Holonomy (whole-part relationship) [illegible] = [illegible] Meronomy [illegible] Holonomy [illegible] =
3. **Hypernym and Hyponym:** are supertype of word and a sub type of word. [illegible] Hypernym [illegible] Hyponymy [illegible]
4. **Toponym** y (cause and effect relationship) [illegible]
5. **Key:** 0- Synonym, 1-Homonym 2-Meronomy, 3- Homonym, 4- Hypernym, 5-Hyponym, 6-Troponymy, 7 - Single-sense, 8 -incorrect grammar

[illegible] !!!

Appendix C: Inter Annotation Agreement progress on WebAnno



Documents	curation	anno1	anno2
last access	13/08/2022 20:06:02	13/08/2022 19:43:09	13/08/2022 19:23:49
ልክ.txt	✓	✓	✓
መንገድ.txt	✓	✓	✓
ሳለ.txt	✓	✓	✓
አካል.txt	✓	✓	✓
ዋና.txt	✓	✓	✓
ዋጋ.txt	✓	✓	✓

Appendix D: Sample ambiguous words with their possible senses

Word Class	word	Possible sense	Number of senses
Nouns	ሀይል	1.ብርታት፣ጥንካሬ፣ጉልበት 2.ቡድን 3.ግፊት 4.አቅም	4
	ሁነኛ	1.ጠቃሚ 2.ተስማሚ 3.ታማኝ 4.ወዳጅ	4
	ሀቅ	1.እወነት፣ትክክል 2.የራስ 3.የተረጋገጠ 4.ልክ፣መጠን 5.ድርሻ	5
	ቀና	1.ቀጥታ፣እወነት 2.ሰመረ፣ለማ 3.አላቀረቅርረም 4.ለመደ፣ተማረ 5.አገሩ ገበረ 6.መቅናት 7.ሰረፀ፣ገባ 8.ኮሶ አሻረ	8
	መለሰ	1.አስታወቀ፣አስረዳ፣ምላሽ ሰጠ 2.ወደኋላ 3.ታረቀ 4.ተወጣ፣ 5.ቀላቀለ፣ደባለቀ 6.አስተካከለ	6
	በራ	1.መላጣ 2.ብርሀን፣ተብቦገቦገ፣ተለኮሰ 3.አቆመ 4.አማረ	4
	ደረሰ	1.ፃፈ 2.መጣ 3.ሄደ 4.ገባ 5.ሴት አወቀ 6.ጨረሰ፣ፈፀመ 7.ቅኔ ተቀኘ	7
Verb	ለቀመ	1.አነሳ፣ሰበሰበ 2.አረመ፣ነቀሰ 3.አጠፋ 4.ጨረሰ 5.ተማረ፣ለመደ 6.ለየ	7

		7.በላ	
	ሰበቀ	1.ጦር ነቀነቀ፣ወዘወዘ 2.አማሰለ፣በጠበጠ 3.አሸ 4.አጣላ፣ነገር ሰራ 5.አጋደመ፣አስተኛ 6.ነከረ	6
	ላፈ	1.ሰለበ 2.አቃጠለ፣ፈጀ 3.ገፈፈ፣አራቆተ 4.ነጠቀ፣ሞጨለፈ	4
	ነጠቀ	1.አወጣ 2.ተማረ፣አወቀ 3.አፈጠነ 4.ወሰደ፣ቀማ 5.ተማረ፣ለመደ 6.አዳነ	6
	ላጠ	1.ቀረፈ፣ገፈፈ፣ገለፈፈ 2.ላጨ 3.ጨረሰ፣አዋገደ 4.መታ፣ገረፈ 5.አዋረደ	5
	ሰራ	1.ጠገነ፣አስተካከለ 2.አደረገ፣አከናወነ 3.ተስተካከለ 4.አዘጋጀ 5.ገነባ	5
Adverb	ገና	1.በአል 2.አልደረሰም 3.አልተፈፀመም 4.ቀድሞ 5.ዕሩር 6.ገነነ	6
	ሰርክ	1.ዘወትር 2.ምሽት	2
	ቀድሞ	1.ፊት 2.ድሮ	2
Adjective	ሃጅ	1. ኢስላማዊ ጉዞ 2.የሚሄድ፣ተጓዥ	2
	ሞገድ	1. ማዕበል፣ሀይል 2.እልህ	2
	ትኩስ	1. ሙቅ፣ያልበረደ 2.አዲስ	2
	ደማቅ	1. የሞቀ፣ያሸበረቀ 2.የጎላ	2