BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF RESEARCH AND GRADUATE STUDIES

FACULTY OF COMPUTING

M.Sc. Thesis on:

Automatic Amharic Word Sense Disambiguation at Sentence Level

By:

Dereje Senay Merawi

Advisor: Tesfa Tegegne(PhD)

June 2021

Bahir Dar, Ethiopia

Automatic Amharic Word Sense Disambiguation at Sentence Level

Dereje Senay Merawi

A Thesis Submitted to the School of Research and Graduate Studies of Bahir Dar Institute of Technology, in Partial Fulfillment for the Degree of Master of Science in Information Technology

Advisor Name: Tesfa Tegegne (PhD)

June 2021

Bahir Dar, Ethiopia

Declaration

I, the undersigned, declare that the thesis comprises my own work. In compliance with internationally accepted practices, I have acknowledged and properly cited all materials used in this work. I understand that non-adherence to the principles of academic honesty and integrity, misrepresentation/ fabrication of any idea/data/fact/source will constitute sufficient ground for disciplinary action by the University and can also evoke penal action from the sources which have not been properly cited or acknowledged.

Name of the student: Dereje Senay Merawi Signature: ______________________

Date of submission: ____

Place: Bahir Dar

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

Advisor Name: Tesfa Tegegne (PhD)

Advisor's Signature: ____________________________

Dedication

In loving memory always given to my Father, ***Mr. Senay Merawi Biru***, because of he died in the age of in his youth. May God rest his soul in paradise.

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

SCHOOL OF RESEARCH AND GRADUATE STUDIES

COMPUTING FACULTY

Approval of Thesis for defense

I hereby certify that I have supervised, read, and evaluated this thesis titled “Automatic Amharic Word Sense Disambiguation model” prepared by Dereje Senay under my guidance. I recommend the thesis to be submitted for oral defense.

Tesfa Tegegne(PhD)	_________________	______________
Advisor’s name	Signature	Date

Acknowledgment

Firstly, I gratefully express my deepest thanks to the almighty God and St. Mary for everything throughout my life and for helping me to reach for today.

I would like to thank my advisor Tesfa Tegegne (PhD) for his great advice, guide, patience, fast response in communication, and strong comments as well as positive encouragement to do this thesis.

I would like to thank Dr. Gashawu Arutie Asaye (PhD in Theoretical and Descriptive linguistic) Assistant Professor at Bahir Dar University for giving his valuable time and knowledge in reviewing our annotated dataset.

The last but not the least, I would say thanks to Mr. Mieraf Mulugeta, computer science instructor at Bahir Dar University Institute of Technology for sharing ideas without limit and positive Encouragement in the area and I also thank my family for their precious love and comprehensive support.

List of Abbreviations

NLP:	Natural Language Preprocessing
Adam:	Adaptive Moment Estimation
WSD:	Word Sense Disambiguation
LSTM:	Long Short-Term Memory
CNN:	Convolutional Neural Network
Bi-LSTM:	Bidirectional Long Short-Term Memory
BERT:	Bidirectional Encoder Representation from Transformers
RMSProp**:**	Root Mean Square Propagation
Glove:	Gloss vector
MT:	Machine Translation
IR:	Information Retrieval
RQ:	Research Question
ELMO:	Embeddings from Language Models
ICAST**:**	International Conference on Advancements of Science and Technology
GPT:	Generative Pre-trained Transformer
Tanh:	Hyperbolic Tangent
ReLU:	Rectified Linear Units
GELU:	Gaussian Error Linear Unit
ELU:	Exponential Linear Unit
ISRLU:	Inverse Square Root Linear Unit

SG:	Stochastic Gradient
SGD:	Stochastic Gradient Descent
MSE:	Mean Square Error
NAdam:	Nesterov Accelerated Adaptive Moment Estimation
Adadelta:	Adaptive Delta
Adagrad:	Adaptive Gradient Descent
Adamax:	Adaptive Moment Estimation Max
SERA:	System for Ethiopic Representation of ASCII format

Table of contents

List of Figures

List of Tables

List of Algorithms

List of Equations

Abstract

Where a word has multiple meanings in a sentence, word sense disambiguation is the method of deciding the correct meaning in a context. This problem occurs at the semantic level of linguistic analysis. Many types of research done in the area especially full resource languages like English rather than Amharic. Some scholars did Amharic word sense disambiguation system in the previous time up to date. However, the most recent scholars didn't consider antonym, troponomy, holonomy, and homonym relations in the WordNet; this problem was solved by this research.

We are designed an automatic Amharic word sense disambiguation model by using a deep learning approach. Since Princeton English WordNet is developed by manual-based WordNet parallelly; we have developed WordNet by using Manual WordNet development strategy. Even if time took but it is an effective method of WordNet development rather than semi-automatic and automatic WordNet development. The prepared manual WordNet contained ambiguous words with its relations and synsets and the corpus of this thesis which has an ambiguous and single sense sentence used as a dataset. Finally, our model disambiguating the given sentence by predicting its relations and fetching ambiguous words gloss from WordNet. The implementation was done on three different deep learning algorithms within three different experiments by 159 ambiguous words, 460 synsets, and 2164 sentences dataset. Overall performance of the model measured by performance metrics like accuracy, precision, F-measure, and confusion matrix using the given dataset. Based on performance measurement LSTM, CNN, and Bi-LSTM achieved 88%, 89%, and 93% accuracy respectively. Therefore Bi-LSTM model has achieved state-of-the-art results related to other models. Experimental results demonstrate that the proposed Bi-LSTM model is effective solving Amharic word sense disambiguation problem.

Keyword: *Word Sense Disambiguation, Deep Learning, natural language preprocessing, WordNet, Senses*

Chapter One

1 Introduction

1.1 Background of the study

Natural language preprocessing is the essential part of computer science in which computational linguistics and machine learning are broadly used. The Machine learns the syntax and semantics of human language; process and it gives the output to the user. The area of NLP involves making computer systems perform meaningful tasks with natural and human-understandable language (Heo et al., 2020).

Kågebäck and Salomonsson (2016)in natural language processing, word sense disambiguation is a challenging task that entails deciding the correct meaning of an ambiguous word in a given context. In WSD, there are two approaches or mainstreams. Labeled meaning is primarily used in supervised methods to train a classifier that produces the proper probability distribution of word senses. Meanwhile, to find the correct word meaning, knowledge-based (unsupervised) methods that specialize in glosses (word sense definitions) often measure the similarity of the context-gloss pair as a ranking.

Kumar et al. (2019)define word sense disambiguation as an important task in Natural Language Processing. The task-related in most cases a word with its correct sense, where the set of any possible senses or contexts within a given window size. The task of WSD was taken as an intrinsic evaluation standard for the semantics learned by sentences in comprehension models. It is an open problem and long history of research.

Word Sense Disambiguation(Alkhatlan et al., 2018) is the ability to automatically determine the correct sense or meaning of a word in context or a given word that has more than one meaning, which one is used in a specific context?. Word sense disambiguation has great attention in English rather than low resource language like Amharic nowadays. Because of the availability of resources in English and Amharic complex structure of morphology. However, the global importance of Amharic should provide motivate or enforce additional in-depth studies. The availability of tools and systems in other languages dramatically affects the performance of the downstream system in Amharic.

Based on (Pesaranghader et al., 2018) WSD is a crucial problem in natural language processing both in its title and as a stepping stone to other advanced tasks within the NLP pipeline, applications such as machine translation. and question answering. When a word is introduced during a brief narrative (surrounding text) that is normally listed as meaning, WSD primarily deals with determining the correct sense of that word among a group of provided candidate senses for that word.

For example, the word ለመለመ(lemeleme) in the surrounding context "የበልግ ወቅት መምጣቱን ተከትሎ የዛፉ ቅጠሎች ለመለሙ"/yebeligi wek'iti memit'atuni teketilo yezafu k'it'elochi lemelemu/ and(ሃረገወይን, እና ሌሎች 1993) "ለመለመ ማለት የዛፍን የአትክልትን ቅጠል በብዛት ቆረጠ ማለትም ነው" /lemeleme maleti yezafini ye'ātikilitini k'it'eli bebizati k'oret'e maletimi newi/.Word Sense Disambiguation is the task of identifying the right sense or meaning of the word or phrase concerning its context. Native speakers of the language disambiguation multiple sense of a word or phrase easily. But the same task becomes highly complex and knotty when done by machines (Abid et al., 2018). It is useful for many applications such as event extraction, document indexing, information retrieval, theme extraction, machine translation, semantic annotation, cross-document referencing, and semantic web search.

In the field of natural language processing, WSD is a great problem, it refers to a task that determines a reasonable sense of a word that can have multiple meanings or seems to be ambiguous in a given context (Heo et al., 2020). It's identifying which sense of a word context is used in a sentence when the word has multiple meanings. This problem occurs at the semantic level of linguistic analysis and a long-standing problem of NLP.

Formally, given a word in context, the task of WSD consists of selecting the intended meaning or sense from a pre-defined set of senses for that word defined by a sense inventory for example the word" ሸመደደ"/shemedede/ based on (ሃረገወይን እና ሌሎች 1993) have senses of "ጠጣ"/t'et'a/ and "አጠና"/āt'ena/ or its glosses "ከጥማቱ የተነሳ የብርጭቆዉን ውሃ ሸምድዶ ጠጣ"/ket'imatu yetenesa yebirich'ik'owuni wiha shemidido t'et'a/ and "የመጻፉን ሁሉንም ገጾች በቃሉ ሸመደደ"/yemets'afuni hulunimi gets'ochi bek'alu shemedede/. We can take also the other example word ሀረግ/hāregi/ has three synsets, three glosses, and relations of the word.

ሀረግ1: ከአጠገቡ በሚገኙ ዛፎችና ቋሚ ነገሮች ላይ እየተጠመጠመ እና እየተሳበ የሚያድግ እንደ ሃረግ ሬሳ፣ ጥሮ አይነት እጽዋት።/ke'āt'egebu bemīgenyu zafochina k'wamī negerochi layi iyetet'emet'eme ina iyetesabe yemīyadigi inide haregi rēsa፣ t'iro āyineti its'iwati/

ሀረግ2: ለጌጥ ሲባል በጥርብ እንጨት፣ በመጽሃፍትና ሐዉልቶች ላይ እንደ ሃረግ እየተለጣጠፈ የሚሰራ ንድፍ። /legēt'i sībali bet'iribi inich'eti፣ bemets'ihafitina ḥāwulitochi layi inide haregi iyetelet'at'efe yemīsera nidifi/

ሀረግ3: ሁለትና ከሁለት በላይ ቃላቶችን የያዘ ማሰሪያ አንቀጽ የሌለዉ የአረፍተ ነገር ክፋይ።/huletina kehuleti belayi k'alatocheni yeyaze maserīya ānik'ets'i yelēlewu ye'ārefite negeri kefayi./

Its relation used as a dataset for this research was

Hypernyms: እጽዋት በዛፎች ላይ የሚጠመጠሙ ሃረጎችንም ይይዛል።/its'iwati bezafochi layi yemīt'emet'emu haregochinimi yiyizali/

Hyponymy: ሃረግ በዛፎች ላይ የሚጠመጠም የእጽዋት አይነት ነው።/haregi bezafochi layi yemīt'emet'emi ye'its'iwati āyineti newi/

As shown above example hypernymy “እጽዋት”/its'iwati/ hyponymy “ሃረግ”/haregi/the family of haregi is its'iwati.

It is a normal task to use their language when accessing information from the internet and using computers. This necessitates the use of a wide range of applications, such as local language spell-checkers, word processors, computer translation systems, word sense disambiguation, and search engines. Question answering, information retrieval, and other NLP tasks benefit from word sense disambiguation.

WordNet (Mulat, 2020)is one of the most commonly used tools in natural language processing for languages like English, however, there were no many resources in Amharic. Word meaning disambiguation necessitates the use of WordNet tools to solve complex problems and improve system performance. Word sense disambiguation is a significant problem at the phonological, lexical, syntactic, semantic level of natural language processing. Disambiguation of Amharic sentences by using WordNet is still a big problem. (Huang et al., 2019)Which aims to find the exact sense of an ambiguous word in a particular context by using lexical resources is a complex task. Besides, Amharic language by nature differs from other languages by morphological variant, and a word to have unlimited derivation and inflection makes WSD difficult.

Knowledge-based, supervised, and semi-supervised approaches were used to solve WSD problems up to recent times (Wiedemann et al., 2019). To infer senses, information-based systems use language tools such as dictionaries, thesauri, and knowledge graphs. On the other hand, supervised methods use an annotated training dataset to train a computer to predict a meaning given the target word and its context. To improve WSD efficiency, semi-supervised approaches to Word Sense Disambiguation combine manually generated training sets with large corpora of unlabeled data. To predict the correct meaning, all approaches depend on some kind of context representation. Context is generally expressed as a function vector obtained from a machine learning model, or as a dictionary resource linked to senses. There were two methods to disambiguate Amharic sentences without WordNet by using context-based models or by using WordNet(Du et al., 2019). This research followed Amharic word sense disambiguation model by using WordNet resources.

To disambiguate a given sentence either with WordNet or without WordNet by contextual embedding. The mechanisms of WordNet development is either manual, semi-automatic, and automatic. The semi-automated method of WordNet development starts with manually created simple lexical databases and then gone to extracting conceptual information from tools like English WordNet, bilingual dictionaries, and monolingual text corpora. Besides this study was followed a manual WordNet development strategy. Because of its effectiveness to disambiguate the intended meaning of the word in the context even if time taker during preparation. Finding Ambiguous words in different sources and give its gloss for those words. The ambiguous word 8 relations within its ID number specified. Create a relationship between ambiguous words, their gloss, and relations. And then build our deep learning model that could understand the words meaning within its surrounding context by using annotated dataset.

1.2 Motivation

Since the Amharic language is useful for every sector in Ethiopia and to some extent outside the country; many Amharic documents are being produced and stored. Amharic word sense disambiguation system by using WordNet is an important component to enhance the performance of Amharic NLP applications. Nowadays, machines learn the Amharic language syntax and semantics within morphological variant and inflection. Robot Sophia is the one whose citizen in Saudi Arabia in the area of AI. She communicates with our prime minister in the Amharic language in previous times. This is the great motive to specialize in the area and want to know how machines understand language especially Amharic?

1.3 Statement of the Problem

Disambiguation is a very difficult task especially low resource languages like Amharic because of the unavailability of resources like WordNet and its complex syntax even if a lot of researchers done on the Amharic language most resources were inaccessible like that of other languages. Besides automatic Amharic WSD is needed to solve such problems.

Different scholars' study on Amharic word sense disambiguation system at different times. (M. Solomon, 2010)attempted Amharic WSD by using supervised manually annotated training data. However, supervised learning methods had a lack of sufficient labeled training data that require expertise.

(A. Solomon, 2011; Wassie et al., 2014)conducted their research on Amharic word sense disambiguation model unsupervised within annotated data, and semi-supervised approach respectively. The experiments that involve instances of a target word are used to learn the context in which the target word is used. Due to the lack of human involvement in mark annotation, unsupervised methods often produce collections of word senses that are not intuitive to humans (Wassie et al., 2014). Supervised approaches, on the other hand, necessitate marking the training data, which is expensive, subjective, and time-consuming. The study considered only verb class ambiguous words and their senses, which makes performance evaluation difficult.

Later, Amharic WSD was done by (Dureti, 2017; Yesuf & Assabie, 2017)a knowledge-based, generic approach respectively. WordNet is constructed as a relational database by considering a maximum of three senses for a word and implemented using the lesk algorithm. But the algorithm includes multiple words having multiple senses are considered at once (combinatorial explosion problem).

Most of the previous Amharic WSDs focused on verb class; however,(Mieraf, 2020) attempted all (verb, noun, adverb, adjective) classes by developing WordNet for Amharic word sense disambiguation. This work also includes synonym, hypernymy, hyponymy, and to some extent meronomy relations of a word. The study implemented semantic similarity by a knowledge-based approach. However, the semantic similarity method had a problem of two concepts that have the same path length have similar distances; even though those have different meanings. Besides, it classifies words that are similar in meaning, but they are not lexicographically similar words in one problem. It's difficult to apply augmented semantic space highly dependent on the number of words since the researchers used 17 ambiguous words.

The Context-to-gloss overlap method was implemented in the study. However, it is only effective for short sentences due to short sentences have a smaller number of context words which makes the frequency count smaller. Since dictionary glosses are typically short and lack of adequate vocabulary to make fine-grained distinctions in relatedness, this was a major restriction. The counter-based approach, such as semantic similarity, would be impractical to implement because it requires significantly more memory and is unable to update the model with new text corpora without retraining from scratch. Gloss vector a traditional word embedding approach(Alkhatlan et al., 2018) have the shortcoming of a single vector representation of a word's meaning, it leads to under control of polysemous words.

So far different methods that apply to WSD have been explored; supervised, unsupervised, knowledge-based, and also semi-supervised methods. Toruno et al. (2020) Supervised methods have better performance related to unsupervised and knowledge-based. However, a significant amount of training data was needed. Recently neural networks have become a dramatic change and fascinating accuracy improvement in the area of NLP. It doesn't require defining features explicitly; instead, it aims to learn different representations from data automatically. That is why we look at neural networks to solve Amharic ambiguous words problem.

To solve the above problems and meet the desired objective of the research, we are designed WordNet to disambiguate the given sentences by a deep learning approach. The study included antonyms, homonyms, troponym, and holonomy relations rather than previous works on the verb, adverb, adjective, and adverb class.

Research Questions

This study attempted to answer the following research questions.

RQ1: How to develop Amharic WordNet?

RQ2: How To develop an automatic WSD model that can disambiguate Amharic ambiguous words in a given sentence?

RQ3: How to test and evaluate a system using a dataset?

RQ4: To what extent Bi-LSTM network disambiguate Amharic ambiguous words?

1.4 The objective of the Study

1.4.1 Specific Objective

To achieve the general objective, we designed the following specific objectives.

- To prepare a dataset with its relation in Amharic ambiguous words.
- To review related works on word sense disambiguation and literature reviews.
- To design Amharic WordNet.
- To design an automatic word sense disambiguation model.
- To test and evaluate Bi-SLTM model using prepared dataset.

1.5 Methodology

In this study, we would follow the design science research approach. Design science has six key steps. Those are problem identification, motivation, objectives of the solution, design and development, demonstration, evaluation, and communication. This research would use six design science steps to achieve the desired objective. The research follows the following steps in different phases.

1.5.1 Problem Identification and Motivation

Problem identification defines the specific problem that has to be resolved and recommended with research work since it is the first step of the research. Within problem identification, the researchers understood the problem detail and helped by the domain Experts during problem identification. The researchers also used domain experts when the time of exploring the problem of that particular area and the way how come up with the respected solution within agreements.

1.5.2 Define Objectives of the Solution

The objective of the study is defined based on the first step of the research problem identification. To construct the objective of the research the problem must be identified and understood by the researchers. It’s supported by reviewing related literature.

1.5.3 Design and Development

Creating the artifact is the third step of research design. Such artifacts are potentially constructing, models, methods, architecture design, dataset preparation, and implementation of algorithms that solve the problem. After design the next step the use of the artifact to solve one or more instances of the problem. The design science research methodology includes specifying the following points in the research.

Literature Review: A literature review would be conducted on various topics related to this research project throughout the thesis life cycle. To demonstrate the current state of the art in this thesis, various approaches to word sense disambiguation would be examined. A comprehensive literature review on word sense disambiguation would be conducted to gain understanding of the topic or knowledge, to identify useful methods for word sense disambiguation, and to improve Amharic WordNet Structure, Amharic ambiguous words, Amharic writing method, and current word sense disambiguation approach.

Tool Selection: For demonstrations, we used python programming software because it's very cross-platform and applicable in a wide range of NLP applications. It integrated with the TensorFlow module within anaconda navigator to develop a model with Keras as the backend. Keras is an open-source library that provides a Python interface for neural networks. It acts as an interface for the TensorFlow library and focused on being user-friendly, modular, and extensible and it enables experimentation with deep neural networks. TensorFlow Google's open-source AI framework for machine learning and high-performance numerical computation in deep learning. It is a free and open-source Python library for machine learning and can execute data-flow graphs. It is used for many tasks but it has particularly focused on training and interface of deep recurrent neural network. We used Microsoft access were one of Microsoft suit application software for Amharic WordNet development.

Dataset Preparation: To achieve the objective of the research dataset preparation is the main task in this study. The data would be collected from online sources sketch engines, Amharic Dictionaries, and domain experts were the data source of this research.

Preprocessing: We applied different data preprocessing techniques on our dataset before data manipulation like tokenization, punctuation removal, stop word removal, normalization, and stemming. Every data before training and testing it must be passed by preprocessing steps.

Algorithms: In this thesis, there were three different deep learning algorithms developed within a specified hyperparameter such as LSTM, CNN, and Bi-LSTM. Those algorithms trained and tested with the train-test dataset and the algorithms which outperform selected for problem-solving. From those Bi-LSTM model achieved state-of-the-art results. Bi-LSTM network composed of forwarding and backward LSTM units to incorporate past and future context information. It can learn long-term dependencies without retaining duplicate context information (Jang et al., 2020). Therefore, it has demonstrated excellent performance for sequential modeling problems, effective encode long-distance word dependencies and text classification. The model handles the central word context by comparing both directions of the word by training backpropagation with time. That is why we used this method to disambiguate Amharic ambiguous words.

1.5.4 Demonstration

The designed Automatic Amharic word sense disambiguation system is demonstrated by using python. Python is an easy to learn and powerful programming language that has efficient high-level data structures. The system implemented on Anaconda 3.7.4 navigator Kera's would develop the model within TensorFlow libraries(environment) as backend. Jupiter was used as a code editor during the experiment.

1.5.5 Evaluation

The evaluation of the Automatic Amharic word sense disambiguation model by using two methods. They used two evaluation metrics used to know the performance of our model. Evaluation metrics and confusion matrix. The model performance was validated based on precision, accuracy, and f1-score with 9*9 confusion matrix result of predicate label and original given dataset. The model which has higher performance selected and used for solving word sense disambiguation problem.

1.5.6 Communication

Communicate the issue and its relevance, the artifact, its usefulness and novelty, design rigor, and effectiveness for researchers and other related audiences and reported here. Following the receipt of the results and conclusions, difficulties, shortcomings, and recommendations are stated. Lastly, at the end of the submission and presentation of this thesis report, in ICAST if it is accepted and we planned to publish in IEEE journal publisher and other journal preceding. To make the research paper general and understood by different people who didn't speak Amharic we append the SERA transliteration for all of the Amharic words in the body of the research taken as examples.

1.6 Scope of the study

The scope of this study would be limited in the disambiguation of ambiguous words in the given corpus with their synonym, hyponymy, hypernymy, meronomy, antonym, holonomy, toponomy, and homonymy of word relations in a given context at sentence level. However, WordNet contains several non-taxonomic relationships that have yet to be exploited by researchers to improve semantic similarity and successful disambiguation. Two types of meronymy relations of a single word: the substance of, and entailment sense disambiguation would not cover in this research because of Amharic WordNet and pre-trained data resource limitation constraint.

1.6.1 General Objective

The general objective of this research was to design an automatic Amharic word sense disambiguation model at sentence level by using a deep learning approach.

1.7 Significance of the study

The result of this study produces experimental evidence that demonstrates the use of bidirectional long short-term memory (BI-LSTM) for the development of an automatic Amharic WSD model. The study contributes a reference in the GitHub repository for future researches and useful for the development of the following NLP application areas.

Machine Translation: WSD is required in MT for words that have different translations with different senses based on context. These senses facilitate and correctly translate the language within its contextual meaning.

Information Retrieval: word sense disambiguation by using WordNet useful for performance improvement of an IR system by using word synonym. Specially, when accessing websites and browse information from search engines to get valuable information easily and regularly needs the deployment of such models.

Speech Processing: WSD useful in the correct phonetization of words in speech synthesis and word segmentation homophone discrimination. In this scenario, the word relation homonymy in the model helps to identify two or more words that have different spelling but the same meaning useful for speech synthesis.

Event Extraction: Obviously, IE means mining information from unstructured text. Wordnets were useful for much information in a different context in the given unstructured corpus. WSD helps to extract/predicate new events that are listed (senses) in the WordNet after training.

Question Answering: WSD is not limited to the listed above application areas but also, it's useful for question answering systems as a knowledge base since users use different languages and a query to answer a single question in search engines.

1.8 Thesis Organization

The rest of this thesis is arranged as follows. Chapter Two deals with the detail of different related works done on Amharic word sense disambiguation and literature reviews on Amharic language morphology, grammatical structure complexity, Amharic ambiguity occurrence, and deep learning approaches especially recurrent neural networks within its new variants like Bi-LSTM internal structure. In Chapter Three, we discussed the design of automatic Amharic word sense disambiguation model details of the design and the proposed system architecture or how the model works and internal structure. In chapter four we discussed experiments of the proposed model on solving the problem, evaluation, result, and discussion of results were analyzed. Finally, in Chapter Five conclusion, contribution, and future work of the research are presented.

Chapter Two

2 Literature Review

2.1 Amharic Language

Amharic is a member of Semitic language in the part of Afro-Asiatic family spoken by most of Ethiopia(ሃረገወይን እና ሌሎች. 1993). The language has its alphabet, called ፊደል/ fidäl, inherited from the Geez is an ancient South Semitic language that now is used only by the Ethiopian Orthodox Tewahedo Church. It has its grammatical structure and its morphological variant pattern that differs from Semitic language and non-Semitic language.

Based on ሃረገወይን እና ሌሎች. (1993) Amharic is one of the families of northern Semitic language and it becomes a great contribution in the area of literature in the 17th century up to the 19th century. Nowadays Amharic language has become one of the known languages in the world and its word size becomes expand day today. That is why we need WordNet resources to disambiguate Amharic words in a given sentence. Based on (Ymam, 2000)Amharic language's base is a sound. When we speak, we mean what is spoken by the tongue during speech. Sound is produced by the tongue during the normal breathing process, in which air from the lungs enters or enters the lungs through a variety of tongues are restrained, released, or moved freely. One also conveys the idea of language by basically creating different sounds in different parts of the tongue.

To improve the effectiveness and efficiency of the developed application, NLP does not require the presence of punctuation marks, no content-bearing words (stop words), or different orthographic representations of the same meaning letters (normalization) in text processing. The same is valid for Amharic language applications such as information retrieval, machine translation, event extraction, question answering, and word sense disambiguation to delete various punctuation marks and common words. The elimination of the punctuation mark and repeated words (stop words) would improve the efficiency of the automatic word sense disambiguation model due to the morphological complexity of the Amharic language.

Amharic words are the linguistically ordered mixture of phonemes and their orthographic representation. Although Amharic has a large number of speakers, writing scripts in the language is difficult. Even though each of the scripts (fidels) has its meaning in the Ethiopian Orthodox Tewahdo Church, in the Amharic language, scripts with identical phonemes but different orthographic representations have the same meaning. As a result, we can interchangeably use identical phonemes with different orthographic representations to return the same value. The scripts (ሀ, ሐ, ኀ), (አ, ዐ), (ጸ, ፀ), (ሰ, ሠ) have the same phonemes to convey similar meaning in the language. For example, the word, "sun" can be written as ጸሀይ, ፀሀይ, ፀሐይ, ጸሐይ, etc. differently. These challenges can be resolved using normalization to work with NLP systems and applications.

The Amharic language lacks uppercase and lowercase letter representations. The written discourse in Amharic contains punctuation marks with various functions that provide the written discourse with the appropriate context for the intended audience. Some of the punctuation mark used in the Amharic language include ሁለት ነጥብ (፡) used for separation of words, አራት ነጥብ **(፡፡)** used for separation of the sentence, ነጠላ ሰረዝ **(፣)** used for separation of Amharic words or phrases with similar concepts, ድርብ ሰረዝ **(፤)** used for separation of Amharic sentences with a similar concept. Some other punctuation marks of Amharic language that are sourced from other languages include ቃለ አጋኖ **(!)** used for making attention to the transmitted information, ጥያቄ ምልክት **(?)** used for determining a request for the situation and wait for a response for the requested information.

The various forms of Amharic words with the inflection and derivation of words make Amharic sentences challenging for Amharic text processing. It has a complex morphological structure with SV/OV/SOV/OSV grammatical structure rather than English and other languages.

Example ልጁ ወደቀ/	**lij**	**u**	**wedek'e**
the boy fell	**Subject**	**suffix**	**Verb**

ኣዳኙ አንበሳውን ገደለው adany u ānibesa wini gedele wi

Sub suffix object suffix verb suffix in the other way it's possible

also አንበሳውን ኣዳኙ ገደለው/ ānibesa wini adany u gedele wi

object suffix subject suffix verb suffix

Besides the syntactic structure of Amharic language complexity, it's difficult to apply other language models like effective pre-trained BERT.

Assabie (2014)after Arabic, Amharic is the second most commonly spoken Semitic language. It employs a one-of-a-kind script known as 'Fidel,' which is written in a seven-column tabular format. The basic form is represented by the first column, and the other orders are derived from it by more or less frequent modifications indicating the various vowels. Amharic has 34 base characters, resulting in a total of 238 = (7*34) Amharic characters. Amharic, like other Semitic languages, has a morphologically complex structure. Its morphology displays a root-pattern phenomenon(Daelemans et al., 2005). A root is a group of consonants also known as radicals whose carries lexical meaning. Vowels are inserted between the consonants of a root to form a stem with a pattern. Specifically, Amharic from Semitic languages verbal stems, consist of a 'root + vowels + template' merger. For instance, the root verb f-r-d+ ee + CVCVC leads to form the stem ferede ('judgment').

2.2 Ambiguities in Amharic Language

Ambiguity is a quality of any thought, idea, statement, or claim whose meaning, intention, or interpretation cannot be determined decisively by a set of rules or processes. Specific and unique interpretations are permitted in ambiguity (though some may not be immediately obvious), whereas it is difficult to create any interpretation at the necessary level of specificity with vague information.

When a word has more than one meaning in a text, lexical ambiguity arises, whereas syntactic ambiguity arises when the order of words has more than one grammatical relationship that changes the meaning of the text. Listeners or readers(Miangah & Khalafi, 2005) can disambiguate ambiguous words in a language that speakers or writers have raised for clarity. Amare (2001) identified six types of ambiguity in Amharic: Lexical Ambiguity, Phonological Ambiguity, Structural Ambiguity, Referential Ambiguity, Semantic Ambiguity, and Orthographic ambiguity. We would summarize those ambiguities as follows with examples.

2.2.1 Phonological Ambiguity

Ambiguity can arise from the interpretation of speech sounds within and across words. Theoretical when the speakers enunciate by making a pause sound, ambiguity ensues. The difference between speaking with and without pauses results in word ambiguity. Structures may be ambiguous because of differences in the placement of pauses within them. For example, the '+' sign shows where the pause is. When they are pronounced without the pause there is a change of meaning. The absence of the pause results in a categorial difference and hence in meaning.

1. ስራ ስሩ ጥሩ ነው = [sira + siru] t'iru newi 'it is good to work'
   The bracketed forms are uttered as a single pause

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

However, if it is pronounced without pause. It will provide different sense from the previous like "it is good work".

2.2.2 Lexical Ambiguity

Words that have several meanings might lead to various interpretations by various people. Individual words or word-level knowledge are the focus of lexical ambiguity(Amare, 2001; Miangah & Khalafi, 2005). When two or more meanings of a term are applicable in the same situation, lexical ambiguity occurs. On the other hand, lexical ambiguity occurs when a lexical unit falls into separate part-of-speech categories with different senses, or when a lexical unit has more than one sense, all of which fall into the same part-of-speech category (Abate & Menzel, 2007). Synonymy, homonymy, and homophonous affixes are some of the causes that might produce lexical ambiguity.

Categorical Ambiguity: it is the first lexical ambiguity and it formed when two lexical items have the same phonological form but belong to different word classes, they are categorically ambiguous. It can be explained using the following ambiguous word:

አክርማ ሰጠችኝ

ākirima set'echinyi - is ambiguous and has the following two different interpretations.

- She gave me akirma (a kind of grass)
- She gave me something after delaying for some time.

ambiguous because / ākirima / could have a nominal or verbal reading. This leads to two different functions as shown in the following deep structures.

እሷ ለኔ አክርማ ሰጠችኝ [+iswa [lenē [ākirima [set'echinyi]]]]

S she VP -I N gave-she-me

[[+iswa [[[+iswa ākirima] [lenē set'echinyi]]]]

S she VP S she make late she V for-I gave-she-me

Homonymy: Another source of lexical ambiguity in Amharic is homonymy. Several lexical elements share the same phonological form but have diverse meanings. Structures like the following suffer from ambiguity as a result of such shapes.

Example bä - wär - e	al– i- f- fata - mm
With-month-me	Neg 1s ps – released – Neg
bä – wäre	al– i- f- fata - mm
with – rumour	Neg 1s ps – released – Neg

'' I will not get frustrated by any rumour'

Homophonous Affixes: Homophonous affixes cause ambiguity in the language. Example

bet-u färräsa has the following two readings

a. The house is destroyed.
b. His house is destroyed.

Because the suffix /- u /can be used as a definite article or a third person masculine identifier, the sentence is ambiguous. The homophonous prefix designating two grammatical functions is where the ambiguity lies.

2.2.3 Structural Ambiguity

In Amharic, structural ambiguity is the most common sort of ambiguity. The term structural here refers to the organization of syntactic elements. When a member of a structure has more than one possible place, structural ambiguity can occur. One surface realization originates from more than one underlying representation in structurally ambiguous statements(Amare, 2001).

የጎጃም ገብስ ጠላ/yegojami gebisi t'ela/

yä-gojjam gebisi t'ela

of- gojjam barely beer readers give the following two interpretations

a) Beer made of barley from Gojjam
b) Beer of barley from Gojjam

2.2.4 Referential Ambiguity

Pronouns need antecedents that are the same number, gender, and person as them. There are two sorts of pronouns in Amharic: free and bound. In the same way that their bound counterparts appear in the verb stems, the free forms may not be phonetically realized in surface structure. For example, ካሳ ስለተመረቀ ተደሰተ - kasa stlä - tä - märräq -ä tädässät -ä

1. Kasa was pleased himself because he graduated. "ካሳ ስለተመረቀ ራሱ ተደሰተ፡፡"
2. Somebody was pleased because Kasa graduated. "ካሳ ስለተመረቀ ተደሰተ።"

2.2.5 Semantic Ambiguity

This type of ambiguity is caused by polysemic, idiomatic, and metaphorical constituents. In Amharic polysemic ambiguity is common.

For example, መብራቱ ጠፋ mebiratu t'efa

a. The light disappeared.
b. The light was off. Or its interpretation

The light went off

Mebratu(a person) disappeared. The constituent / mäbratu / a definite NP may refer to 'the light' or a male person.

2.2.6 Orthographic Ambiguity

In orthographic ambiguity algorithm does not distinguish between geminate and non-geminate sounds, some structures may be problematic for orthographic reasons. The context can be used to deduce a word's intended meaning. However, in other cases, such as the following, this is not possible.

መኪናው ይሰራል- [mekīnawi yiserali] from this yiserali was the source of ambiguity.

a. The car works
b. The car will be repaired

2.3 Overview of Word Sense Disambiguation

Word Sense Disambiguation is a critical problem in Natural Language Processing, both as a standalone problem and as a stepping stone to more advanced NLP tasks like machine translation(Vickrey et al., 2005) and question answering(Hung et al., 2005). When a word is introduced, WSD is concerned with determining the correct meaning of that word from a collection of candidate senses in a brief narrative (surrounding text) which is generally referred to as context(Pesaranghader et al., 2018).

Based on (Yuan et al., 2016) word sense disambiguation is a long-standing issue in natural language processing that involves determining the intended meaning of words in a text. Researchers have recently demonstrated promising results in WSD algorithms using word vectors derived from neural network language model-like features. A simple average or concatenation of word vectors for each word in a document, on the other hand, losses the sequential aspect and syntactic information of the text.

A term is said to be ambiguous if it can be interpreted in several ways, each with a different meaning or sense. This occurs in every natural language's vocabulary, and humans can easily figure out what the ambiguous terms mean by looking at the context in which they appear. Every WSD model task aims to allow machines to understand the correct meaning of these ambiguous terms in the same way that humans do. Any WSD model's protocol involves using a method that uses one or more sources of information to connect the most suitable senses with a series of words in context(Giyanani, 2013).

Kågebäck and Salomonsson (2016)words are inherently vague, and depending on the context, they may have several similar or unrelated meanings. For example, the word rock may refer to both a stone and a musical genre, but the meaning of rock is clear in the sentence "Without the guitar, there would be no rock music". Word sense disambiguation is the process of matching a word token in a text, such as rock, to a well-defined word sense in a lexicon. It's clear from the rock illustration above that the context surrounding the word is what clarifies the meaning. However, the difficulty of the mission may not be immediately apparent. For example, in Amharic words the word ሳለ/sale/ have three different senses in a different context

Synset1: ከጉንፋን ወይም ኮሮና በሽታ የተነሳ ከጉሮሮው አየርን በሃይልና በተደጋጋሚ አስወጣ ኡህ ኡህ አለ። /kegunifani weyimi korona beshita yetenesa kegurorowi āyerini behayilina betedegagamī āsiwet'a uhi uhi āle/

Synset 2: ሞረደ፣ አሾለ፣ አተባ፣ ስለት አወጣ/morede፣ āshole፣ āteba፣ sileti āwet'a/

Synset 3: የአንድን ነገር መልክ፣ ቅርፅ ወይም አንድን አይነት ሀሳብ በስዕል አሳየ/ye'ānidini negeri meliki፣ k'irits'i weyimi ānidini āyinete hāsabi besi'ili āsaye/

2.2 Language Models

Fine-tuning language models is a downstream task contextualized and it's used to simplify plugin the task-specific inputs and outputs into pre-trained models, such as BERT, all part of the parameters are end-to-end(Wiedemann et al., 2019). This procedure adjusts the model's parameters according to the objectives of the target task, like the classification task in WSD. One of the main drawbacks of this type of supervised model is their need for building a model for each word, which is unrealistic in practice for all-words WSD.

Du, Qi, and Sun (2019a)undertaken Embedding from Language Models (ELMO), Generative Pre-trained Transformer (GPT), and Bidirectional Encoder Relational Transformer (BERT) recent pre-trained language models, effective to extract features from plain text in English and Chines languages because of reach resources like SemCor (Semantic annotated Corpora). The pre-trained ELMO embedding is WSD features, but there are no studies that fine-tune language models on WSD. However, it's not implemented in the Amharic language due to a lack of semantically annotated corpora and WordNet resources.

Loureiro et al. (2020)tried Shallow machine learning models that cost lots of time on design features for each task. The recursive neural network can automatically learn the semantics of text recursively and the syntax tree structure without feature design. Many fields in NLP have been swept by transformer-based language models. Because of their ability to capture context-sensitive semantic difference, BERT and its derivatives dominate most current assessment metrics, including those for Word Sense Disambiguation. However, nothing is known about their ability to encode and restore word senses, as well as their limitations.

Word embeddings have certainly been one of the most common topics in lexical semantics research over the last decade. Mikolov et al. (2013)word2vec's appearance as one of the first word embedding models set off a huge wave in the field of lexical semantics, the effects of which can still be felt today. Static word embeddings, on the other hand (such as Word2vec), have the drawback of becoming set or context insensitive. In all cases, the word is associated with the same representation, even though different contexts may cause different meanings of the word, some of which may be semantically unrelated. Sense representations is an attempt to remedy the deficiency of word embeddings in terms of sense conflation.

(Maslej et al. 2020)Neural networks are the best performing deep learning algorithms nowadays. They have brought great success within the field of AI, in this study, we used them for text classification purposes.

2.2.1 Convolutional Neural Network

Convolutional neural networks (CNN) are a type of forwarding neural network that includes a layer of neurons that performs the convolution process(Maslej et al. 2020). The architecture of this network was influenced by the role of the ocular nerve. Neurons respond to the input of surrounding neurons' activations using a convolutional kernel, also known as a filter, of a certain scale. CNNs are deep, feed-forward artificial neural networks successfully applied to analyze visual imagery, Part-of-speech tagging, name object recognition, semantic parsing, sentence modeling, and search query retrieval have all recently been shown to be successful with CNN models(Collobert et al., 2011; Kalchbrenner et al., 2014; Shen et al., 2014).

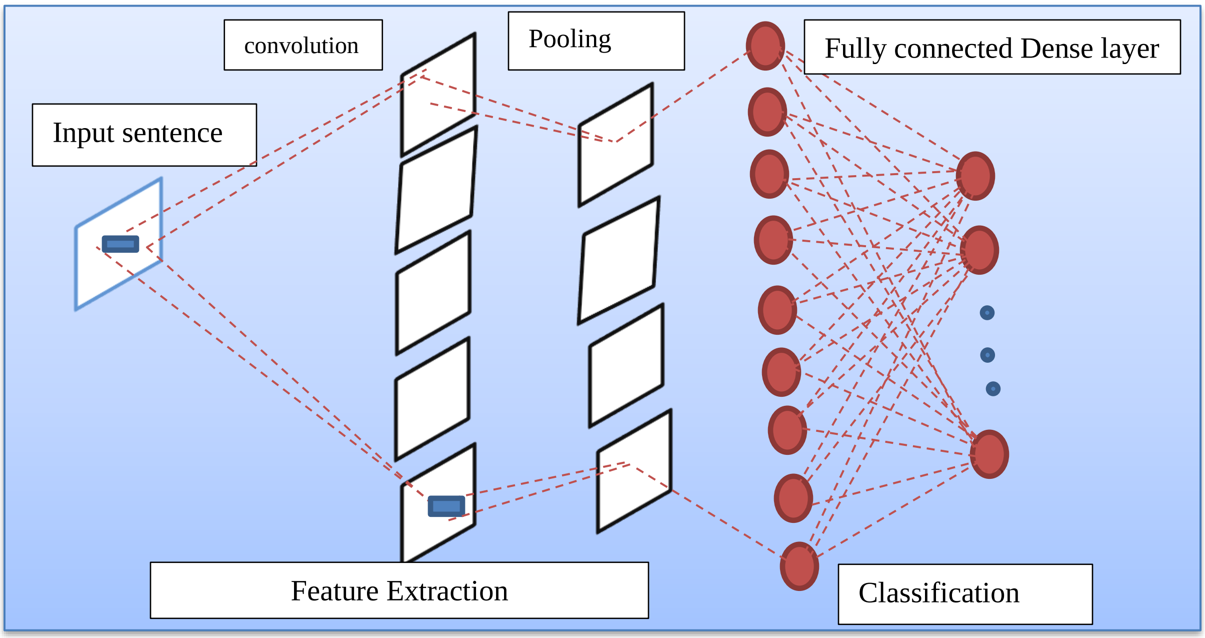


Figure 2:2 Convolutional Neural Network[Maslej et al. 2020]

As shown in figure 2:2 CNN includes three types of layers: convolutional, pooling, and fully connected layers. To compute feature maps from the previous layer, each convolutional layer has many convolution kernels. The convolution method will learn distinct feature representations of original inputs and pass them to subsequent layers. The receptive field of a single neuron is described as a region of neighboring neurons that is mapped to a single neuron on the next layer. The input is convolved by a trainable kernel, and the result is applied with an element-wise nonlinear activation function to produce a feature map. Pooling layers are used to reduce the number of outputs, reduce computational complexity, and avoid over-fitting. Since duplicate data is generated when the convolution layers are applied, the sampling layers are normally applied just behind them. The kernels change as they pass through the individual inputs(Ogunfunmi et al., 2019).

2.2.2 RNN as a Language models

Ogunfunmi et al. (2019)any network with a cycle within its network connections is referred to as a recurrent neural network (RNN). That is any network in which the value of a unit is directly or indirectly determined by the network's previous outputs as an input. Such networks are difficult to reason about and train, despite their strength. However, there are restricted architectures within the general class of recurrent networks that have proven to be extremely successful when applied to spoken and written language.

RNN-based language models go through sequences one word at a time, trying to predict the next word in the sequence using the current word and the previous hidden state as inputs. Because of the minimal context constraint, is avoided in N-gram models because the secret state contains information about all previous terms back to the start of the series. For example, the input sequence x is made up of one-hot vectors of size |V| *1 that represent word embeddings, and the output predictions y, are made up of vectors that represent a probability distribution over the vocabulary. At each step, the model retrieves the current word's embedding using the word embedding matrix E, then combines it with the hidden layer from the previous step to compute a final result and added a new secret layer. After that, this hidden layer is used to produce an output layer, which is then passed through a SoftMax layer to generate a probability distribution for the entire vocabulary(Ogunfunmi et al., 2019).

2.2.3 Long Short-Term Memory

Recurrent neural networks (RNN) have recently shown great promise in a variety of deep learning sequence modeling tasks, such as automatic speech recognition, text classification(Graves, 2012; Lu & Salem, 2017), and language translation. Simple RNNs, on the other hand, have been stated to exhibit the vanishing and exploding gradient problems occurred when trained using stochastic gradient descent(Bengio et al., 1994). When we say vanishing, problem means its cost function doesn’t reach global minima because of slow movement. On the other hand, exploding gradient problem occurred the cost function doesn’t reach the global minima due to its up down at the same place. This has limited the ability of simple RNN to learn sequences with relatively long dependencies.

Researchers have developed a range of techniques in network architectures and optimization algorithms to overcome this limitation(Hochreiter & Schmidhuber, 1997), among which the most successful in applications is the Long Short-term Memory (LSTM) units in RNN. An LSTM unit has a memory cell that can keep its state value for a long time, as well as a gating mechanism with three non-linear gates: an input, an output, and a forget gate. The gates' purpose is to control the flow of signals into and out of the cell, allowing for efficient long-range dependency regulation and good RNN training. Many changes have been made to the LSTM unit since its introduction to enhance performance. Gers, Schraudolph, and Schmidhuber (2002) have added "peephole" connections to the LSTM device, which link the memory cell to the gates and enable precise timing of the outputs to be inferred. Sak, Senior, and Beaufays (2014) between the LSTM units' layer and the output layer, two recurrent and non-recurrent projection layers were added, resulting in significantly improved performance in a broad vocabulary speech recognition mission. Adding more components to the LSTM unit's architecture, on the other hand, can make the learning process more difficult and make it more difficult to understand the function of individual components.

Recently, researchers proposed several simplified variants of the LSTM-based RNN. Cho et al. (2014)Gated Recurrent Unit (GRU) RNN is a two-gate-based architecture in which the input, forget, and output gates are replaced by an update gate and a reset gate. Chung et al. (2014) on the particular dataset used, performance comparisons between LSTM and GRU RNNs were presented, with the latter performing comparably or even better than the former. These findings are still being tested in further studies and with larger datasets. The GRU model found that coupling the input and forget gates and eliminating peephole connections had no major impact on output. They also say that the forget gate and output activation are essential components. These findings were collaborated by the work of (Jozefowicz et al., 2015) who evaluated extensive architectural designs of ten thousand different RNNs.

In(Jozefowicz et al., 2015) the output gate was found to be the least significant when compared to the input and forget gates, so the authors suggested applying a bias of 1 to the forget gate to boost the LSTM RNN's efficiency. Zhou et al. (2016) by combining the update and reset gates in the GRU model, a Minimal Gate Unit (MGU) with a minimum of one gate, namely the forget gate architecture, was proposed. The authors discovered that an RNN with fewer parameters, the MGU model, with the GRU model in terms of testing accuracy. Recently, Salem (2016) introduced a simple method for simplifying the traditional LSTM model by concentrating solely on the generation of gating signals. The gating signals can be used as general control signals, with the loss function/criterion being minimized. All three gating equations were kept, but the parameters were reduced by removing one or two of the signals that drove the gates.

Generally, the Recurrent Neural Network approach is an effective deep neural network it predicates time sequence. However, the longer the sequence, the higher the chance of encountering, vanishing, and exploding gradients. Due to it makes accurate prediction of difficulties. To solve this problem LSTM is presented as a variant structure of RNN. The LSTM consists of a memory cell to save data, an input gate to receive data, an output gate to export data, and a forget gate to delete or restore data(Heo et al., 2020). The internal structure of a single LSTM cell has three parts input, output, and forget gates represented in figure 2.3

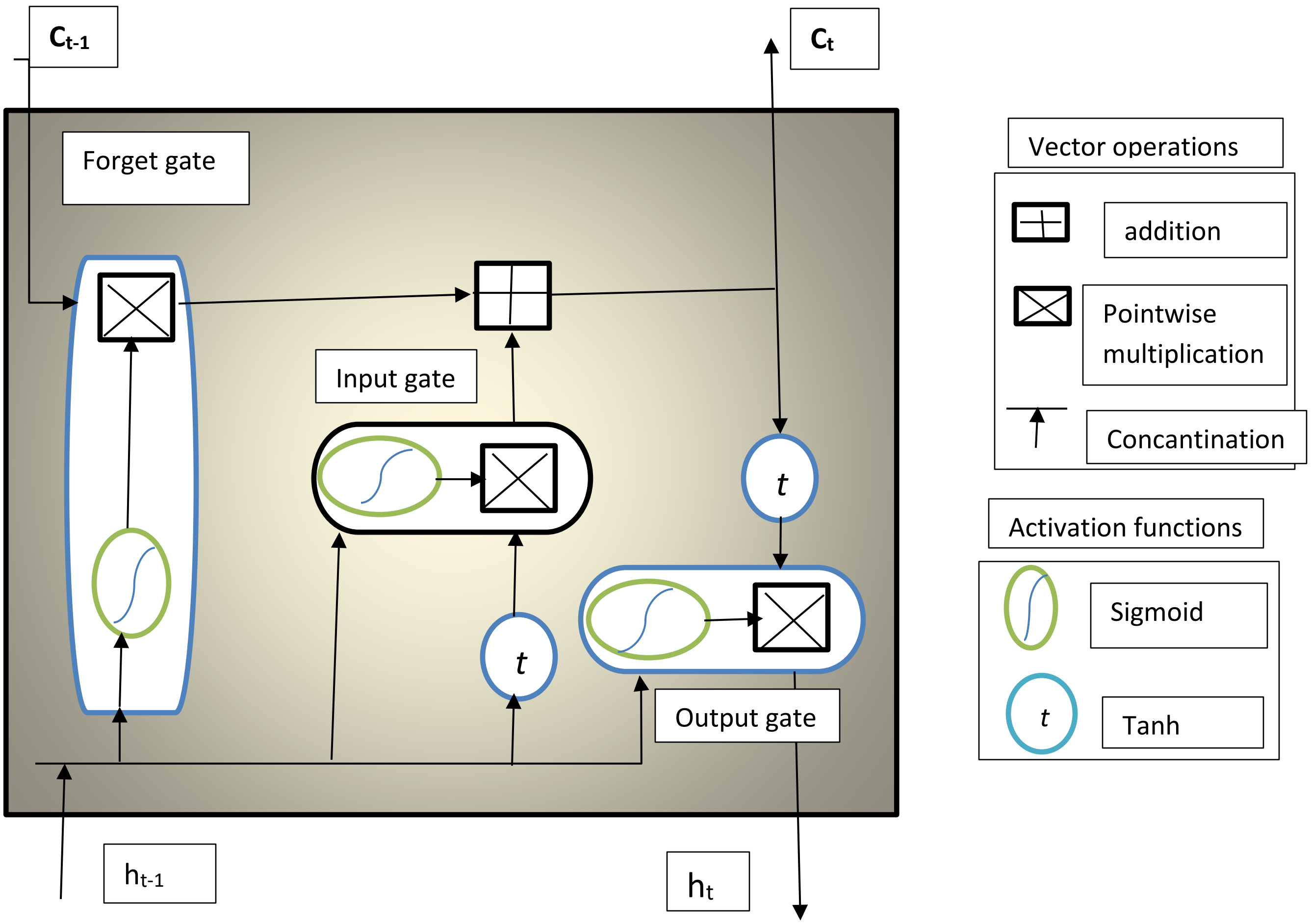


Figure 2:3 Long Short Term Memory Cell Gates[(Kleenankandy & KA, 2020)]

2.2.4 Bidirectional Long Short-Term Memory Model

Bidirectional long short-term memory is a type of recurrent neural network that focuses on sequential knowledge. The information from the previous status of neurons is used in Long Short-Term Memory models(F. Braz et al., 2018). The data processing flows backward and forward at a time and the output of each LSTM is merged using their sum. Besides more contextual information can be extracted using bidirectional LSTM(Bi-LSTM) at a time. These models have more effective in solving the problem of speech recognition, being a state-of-the-art result classifying sequential data into multiple classes. Since our Amharic word sense disambiguation also a multi-class, therefore Bi-LSTM is a suitable model. The general architecture of Bi-LSTM is shown in figure 2.4.

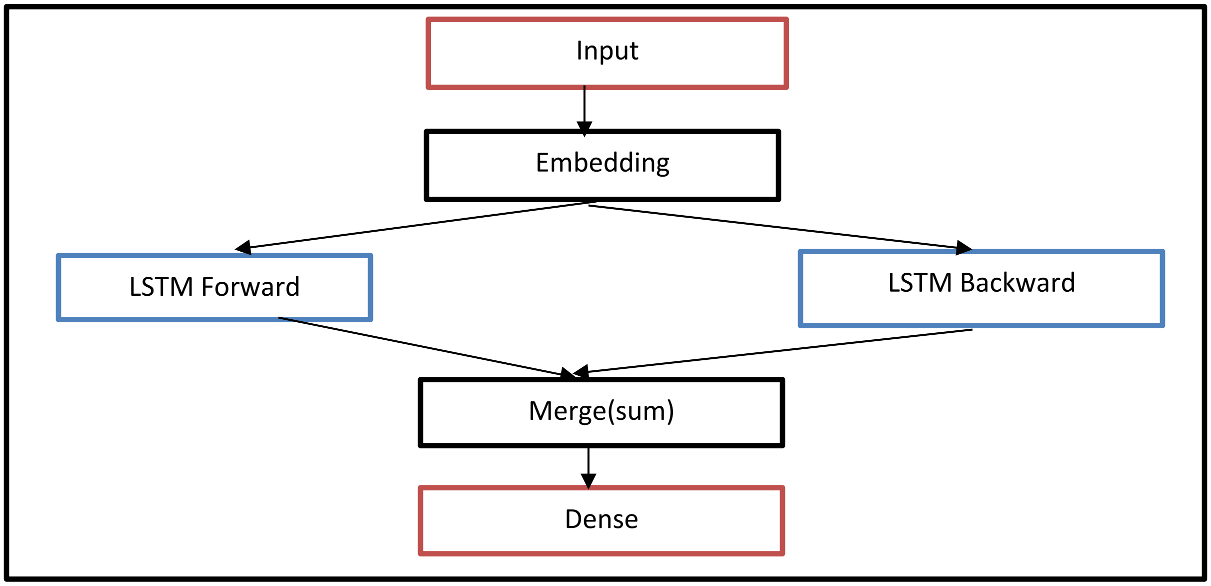


Figure 2:4 Bi-LSTM Architecture diagram [(F. A. Braz et al., 2018)]

The LSTM architecture is a gated recurrent neural network that was created to solve the vanishing and exploding gradient problems that plague traditional RNNs. It has cyclic connections, unlike feedforward neural networks, which makes them useful for modeling sequences. A bidirectional LSTM is made up of two unidirectional LSTMs that have been reversed. For WSD, this means we're ready to encode information from both preceding and subsequent words in the context of an ambiguous expression, which is crucial for correctly classifying its meaning.

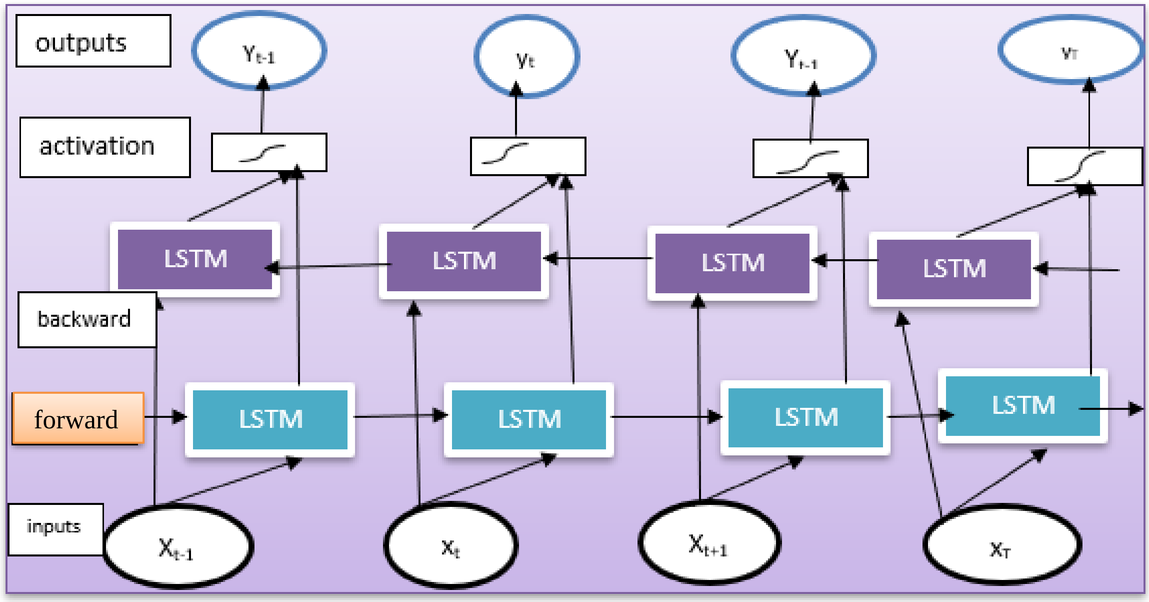


Figure 2:5 Bi-LSTM Internal Structure[(Maslej et al. 2020)]

Based on figure 2:5 shown above Bi-LSTM RNN model have two backward and forward LSTM layers. Every output of precedence LSTM used as input for the successor LSTM and each output of forwarding LSTM and backward layer sum up by activation functions to gave a single output.

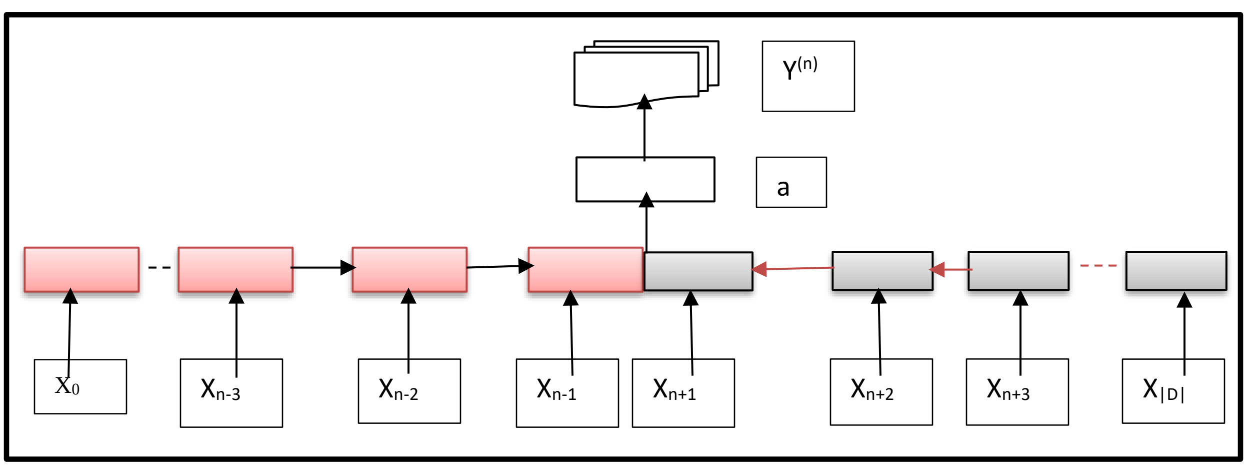


Figure 2:6 A BLSTM centered Architecture around a Word at Position n

As shown in the figure above the Bi-LSTM output is fed to a neural network sense classifier consisting of one hidden layer with linear units and a SoftMax. SoftMax selects the corresponding weight matrix and bias vector for the word at position n. In this type of prediction Bi-LSTM model predicts the context of a word at the center position it compared both directions up to the lengthening of the last two words X_0 up to $X_{|D|}$ since those gone to maximum padding length.

2.2.5 Activation functions

Activation function can be classified into two network architecture and conventional activation. Conventional includes Tanh and Sigmoid activation functions and network architectural ReLU, ELU, SELU, GELU, and ISRLU included(Nguyen et al., 2021). Mapping the product of inputs and trainable parameters was the function of Tanh functions to a range of -1 to 1. The Back Propagation process reveals a potential problem called vanishing gradients, which occurs when the gradient too small or too big, or a saturated problem. The same as Tanh Sigmoid also problem of vanishing gradient approaches to 0 and 1. This problem causes trainable parameters would not to be adjusted, resulting in stopping learning. However, when ReLU is trapped on the left side of zero, it may experience a problem known as a dead state. When a problem affects a large number of nodes in a network, the network's output suffers.

(Nguyen et al., 2021)furthermore, neurons have the same sign of gradient with ReLU, all of the layer's weights either increase or decrease. To solve vanishing gradients and dead state ElU developed. The same as ReLU in positive functions, but differ in negative functions. In negative functions, ReLU responded 0 but ELU produces negative values even if weights updated only in one direction cannot occur problems. SELU is developed for batch normalization before feeding data to the neural network. The above-listed activation function didn’t solve batch dropout regularization, but GELU solved. The exponent function on the left side of zero highly cost computation problem occurred in ELU. To solve this problem and speed up learning ISRLU developed but it needs a different experimental setup.

2.3 Approaches to WSD

Word ambiguity problems are solved by different mechanisms before a time and nowadays. Like, Knowledge-based, corpus-based, hybrid approach that combines both corpus-based and knowledge-based methods and contextual embedding without WordNet were different approaches used for solving WSD problems in different languages in the world.

2.3.1 Knowledge-Based Approaches

Methods that use explicit lexicon, such as MDR (Machine Readable Dictionary), thesauri, ontologies, collocations, and others, to extract information from word meanings and relationships among word senses are known as knowledge-based approaches for WSD(R. Kumar et al., 2012; Navigli, 2009; Shree & Shambhavi, 2015). Researches on WSD before the 1980s and 1990s was theoretically important but realistic in restricted domains before these lexical tools became widely accessible(Shree & Shambhavi, 2015). The first knowledge-based approaches to WSD were developed in the 1970s, but due to a lack of large-scale computational tools, proper evaluation, comparison, and application of these methods were impossible(Navigli, 2009).

Information-based methods depend solely on the above-mentioned knowledge sources and do not use corpus evidence. As a result, these systems are a viable alternative to supervised systems that depend heavily on large amounts of sense-annotated data(Basile et al., 2008; Shree & Shambhavi, 2015). When training data is usable, this makes knowledge-based systems ready to use and scalable, but they achieve lower precision than supervised corpus-based approaches(Basile et al., 2008; Shree & Shambhavi, 2015). This poor performance is due to their complete reliance on dictionary-defined senses, the readiness of which must be ensured, as well as a lack of world awareness(Shree & Shambhavi, 2015). Knowledge-based approaches can be divided into four categories. The overlaps-based method or Lesk algorithm, selection preferences or restrictions, semantic similarity, and heuristics are the types of knowledge-based approach(Chaplot & Bhattacharyya, 2014; R. Kumar et al., 2012; Navigli, 2009).

2.3.2 Corpus-Based WSD Approach

The Corpus-based WSD method uses feature vectors to reflect meaning. Word collocations, POS marks, domain details, grammatical relationships, and other features are included. These methods are then used in a fully automated learning process. However, they are highly reliant on human interaction, the type of training data required, the nature of linguistic expertise employed, and the performance produced (Shree & Shambhavi, 2015). When large amounts of training data are usable, corpus-based algorithms outperform knowledge-based systems. However, this necessitates a large amount of data and creates an information acquisition bottleneck, especially when supervised WSD is used(Pierpaolo et al., 2008). It can classify as supervised corpus-based and unsupervised corpus-based approaches.

Supervised Corpus-Based WSD Approach

This approach trains machine learning algorithms using a corpus of sense-annotated training data(Rajani & Ravi, 2015). This method achieved good results but it needs a large amount of annotated data; the problem encountered in such a situation known as knowledge aquation bottleneck. It needs at least 3700 words in 1000 sentences required for starting such methods to solve a word sense disambiguation problem. It includes any machine learning algorithms like decision list, decision tree, Example-based KNN, SVM, Naive Bayes. The one from the remaining Naive Bayes has the following problems

- It assumes a spherical shape for each sense cluster, being unable to accurately model the decision boundaries given the limited number of examples.
- It has no training data for and does not model, the sense prior, omitting an extremely powerful potential signal. To overcome such problems the scholars designed a semi-supervised word sense disambiguation method.

A supervised WSD program learns the requisite disambiguation information from a broad sense-tagged corpus, in which word occurrences have been manually tagged with senses from some wide-coverage dictionary, such as the LDOCE or WORDNET(Ng & Zelle, 1997).

Unsupervised Corpus-Based WSD Approach

While supervised approaches have the highest accuracy and efficiency, obtaining the necessary resource is difficult and time-consuming (Devendra, 2014). Even if we have sufficient resources, it is difficult to scale for use in other languages, and the disambiguation is limited to a defined number of senses in the repository. Supervised learning is one of the approaches taken to address these issues. Unsupervised learning, on the other hand, is data-driven and language-independent, making it easily scalable to new languages and domains.

Rather than sorting, unsupervised WSD conducts sense grouping (clustering). While it solves a variety of problems associated with unsupervised learning, it is not without flaws. Since it is completely unsupervised, it has poorer output than other methods. Furthermore, the number of clusters and the actual number of senses can vary, making performance assessment difficult. As a result, checking the quality of clusters requires humans to search for relationships between cluster members(Roberto, 2009).

Unsupervised WSD uses a variety of techniques, approaches, and algorithms. The two key approaches are distributional and translational equivalent approaches, which he refers to as knowledge-lean approaches since they depend solely on un-annotated corpora. Discrimination is the focus of distributional methods, which are focused on the idea that terms that appear in similar contexts have similar meanings. Translational equivalent methods, on the other hand, make use of parallel bilingual corpora in two languages that can be used to create a meaning inventory automatically.

Under the mentioned approaches, there are two important methods; token-based and type-based methods(Roberto, 2009). Token-based methods, in the case of a distributional approach, cluster (group) contexts in which the target word appears with the same meaning into one group. Type-based methods, on the other hand, classify and cluster terms that are likely to occur together in related contexts. In the translational equivalent, the form-based method generates a collection of similar terms in the source language (bilingual dictionary), while the token-based method generates meaning-tagged text by providing sufficient translation for each occurrence of the target word.

2.3.3 Semi-supervised WSD Approach

Word sense disambiguation is a long-standing issue in natural language processing that has a wide range of applications. WSD has been researched using supervised, unsupervised, and knowledge-based methods(Navigli, 2009). They presented a semi-supervised approach that augments the labeled example sentences with a large number of unlabeled sentences from the web to resolve supervised WSD approach drawbacks. From the labeled to the unlabeled sentences, sense labels are propagated. The judgment boundary between different senses can be better approximated by adding a large number of unlabeled sentences.

2.4 WordNet

WordNet functions similarly to a dictionary it stores words and their definitions. However, vary from conventional ones in several respects. Words in WordNet, are organized semantically rather than alphabetically. Synonym sets, or synsets, are collections of terms that are synonyms. Each synset corresponds to a single meaning or definition. The most basic item in WordNet is a synset, which is a collection of strict synonyms. Each synset in which a word appears is, by definition, a different meaning of that word. WordNet is a lexical database with a wide number of entries. Cognitive synonyms (synsets) are groups of nouns, verbs, adjectives, and adverbs that each express a distinct meaning. Synsets are linked through conceptual-semantic and lexical relationships.

Fellbaum (2010)WordNet is broad, it could be built manually, semi-automatically, and automatically built a semantic network that connects words with similar meanings. Although WordNet's goal is no longer to model human semantic organization, it has become a key tool for Natural Language Processing and has sparked research in lexical semantics and ontology. The architecture and electronic format of WordNet have proven useful in a variety of Natural Language Processing (NLP) applications, including mono-lingual and cross-linguistic information retrieval, question-answering systems, and machine translations. Word sense recognition is a difficulty in both of these activities due to lexical polysemy. In certain instances, statistical methods can define the context-intended meaning, but they have limitations. WordNet enables automated systems to identify and quantify the semantic relatedness of polysemous terms that co-occur, facilitating alternative or complementary symbolic approaches to word sense discrimination. The semantics-based framework of WordNet allows for targeted look-up of meaning-related terms and concepts from a variety of sources. The arcs between WordNet's words and synsets, unlike in a conventional thesaurus like Roget's, convey a finite number of well-defined and labeled connections.

For example, in our WordNet an ambiguous word ሰፈረ/sefere/ have three senses ለካ/leka/, አመቻችቶ ቤት ሰራ/āmechacheto bēti sera/, and አንቸረፈፈ/ānicherefefe/ and 8 relations joined by many-to-many relationship with its gloss. The synsets of this ambiguous word

ሰፈረ1፡ ለካ መጠነ ለኬቱን ማወቅ ለምሳሌ በክንድ ሰፈረ/leka met’ene lekētuni mawek’i lemisalē bekinidi sefere/

ሰፈረ2፡ ከአፍ እስከገደፉ ጢም አድርጎ ሞላ/ke’āfi isikegedefu t’īmi ādirigo mola/

ሰፈረ3፡ መኖሪያ ቦታ ያዘ፣ ነዋሪ ሆነ ቆጥ ላይ ወጣ /menorīya bota yaze፣ newarī hone k’ot’i layi wet’a/

2.4.1 Synonym Relations

The term "synonym" has many meanings. Synonym words are words that have almost or the same meaning as another word in some way. The term "synonymy" refers to the existence of terms with closely related meanings. In most cases, a synonym for a group of words may be found in dictionaries or thesaurus, although it is normally incomplete. This is because words do not have a single meaning, it may have many aspects of meanings in different contexts. So it is difficult to judge whether two words are synonyms or not (C. W. Li, 2017). Synonyms have several different meanings depending on the domain. For example, if someone tries to write a document in a word processor and substitute a word with its synonym, he or she would most likely choose something similar to the dictionary meaning of a synonym. Examples of synonym in the dataset sentences

Synonym1 - ቀለመ ማለት ያዘ፣ ነካ በቀለም ተነከረ ከሚሉ ቃላት ጋር ይመሳሰላል/k’eleme maleti yaze፣ neka bek’elemi tenekere kemīlu k’alati gari yimesaselali/ from the sentence the ambiguous word is ቀለመ.

Synonym2 – ambiguous word ሽሪራ with its synonym meaning within the context ሽሪራ የሚለው ኮርኒስ ላይ የሚመታ ቀጭን እንደ ማስመሪያ ያለ እንጨት ጋር ተመሳሳይ ትርጓሜ አለው።/sherīra yemīlewi korinīsi layi yemīmeta k’ech’ini inide masimerīya yale inich’eti gari temesasayi tirigwamē ālewi/

2.4.2 Hypernymy-Hyponymy Relations

(Sahin, 2017)a semantic relationship between a generic and unique word is represented by hyponymy. The general term is called hypernym, and the particular term is called hyponym. Class and sub-class are defined by hypernymy and hyponymy, respectively.

Sub-classes are more specific than this class. Based on the definition, an element is co-related with others in which that contain common words (mutual information) with a minimum cosine distance. The word having minimum summation cosine distance and which is universal for selected synsets is considered as the hypernym (x) of all other lists of words in the synsets. The Hyponymy relationship is often represented by the “X may be a kind of Y” pattern. X and Y represent any hyponym and hypernym word, such as apple-fruit and dog-animal, respectively, in this pattern. Hyponymy is an asymmetrical relationship. Although the condition "every X is a/an Y" is valid, the converse (each Y is a/an X) is not true. As a result, X and Y cannot be substituted for one another. Hyponymy may be a transitive linguistic relationship.

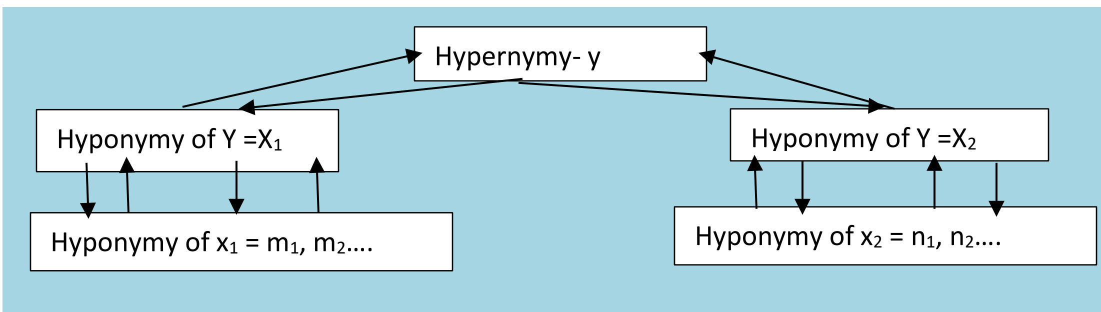


Figure 2:7 Hypernym Hyponym Relation[(Sahin, 2017)]

Examples of Hypernym and Hyponym in the dataset sentences

Hypernym1 -እንስራውን በቤት ውስጥ ከሚዘጋጁ መጠጦች አንዱ በሆነው ጠጅ ሾመ/inisirawini bebēti wisit’i kemīzegaju met’et’ochi ānidu behonewi t’eji shome/- Hypernym of ጠጅ is መጠጥ(family)

Hyponym1 - የተሾመው ጠጅ በቤት ውስጥ ከሚዘጋጁ መጠጦች አንዱ ነው/yeteshomewi t’eji bebēti wisit’i kemīzegaju met’et’ochi ānidu newi/- Hyponym from the ambiguous word ሾመ hyponym is a kind of Hypernym since it’s the largest class and member relationship.

Hypoyomy of መጠጥ is ጠጅ(member)

Hypernym2 - የቤት ኮርኒስ ከእንጨት፣ ከፕላስቲክ እና ከ ሽራሪ ይሰራል።/yebēti korinīsi ke’inich’eti፣ kepilasitīki ina ke shirarī yiserali/-ኮርኒስ(family) is Hypernym of ሽራሪ(member)

Hyponym2 - ሽራሪ ለኮርኒስ መሲርያ ከሚያገላግሉ ቁሳቁሶች ውስጥ አንዱ ነው።/shirarī lekorinīsi mesīriya kemīyagelagilu k’usak’usochi wisit’i ānidu newi/-ሽራሪ(member) is Hyponym of ኮርኒስ(family)

2.4.3 Meronym-Holonomy Relations

Holonomy represents a semantic relationship between a whole term and a partial term. In this relation, part of a whole is called a meronym and the whole of a part is called a holonomy. Holonomy relationship can be represented by "X is part of Y" and "X is member of Y" patterns. In these patterns, X and Y represent any meronym and holonomy term such as wheel-car, leaf-tree, etc., respectively. As in hyponymy, holonomy is asymmetric and transitive linguistic relation. If X may be a meronym of Y and Y may be a meronym of Z, then X may be a meronym of Z. Given two propositions "nail is part of a finger" and "finger is part of the arm", "nail is part of the arm" can be extracted using transitivity(Sahin, 2017).

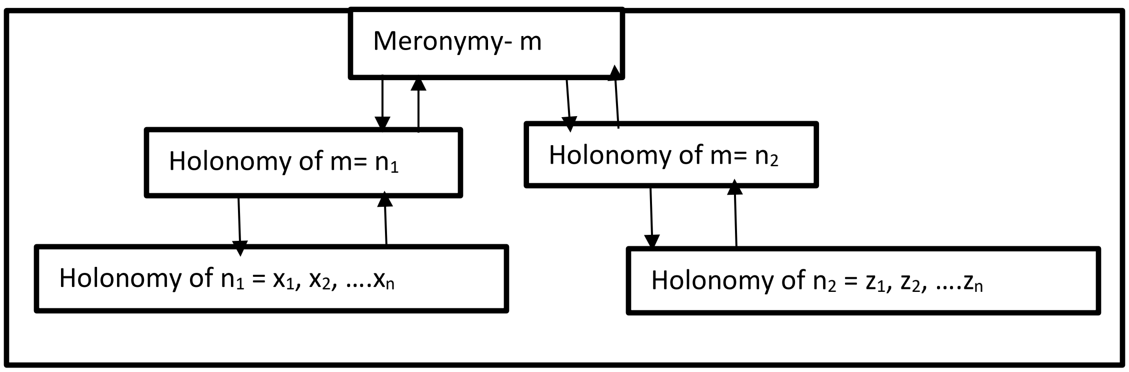


Figure 2:8 Meronymy Holonomy Relation[(Sahin, 2017)]

Examples of meronomy and Holonomy relationship in the dataset sentences.

Meronomy1 - ጠላ የተሾመበት እንስራ ጆሮ ተበጠሰ/t'ela yeteshomebeti inisira joro ebet'ese/Meronomy of ጆሮ(part) is እንስራ(whole)

Holonomy1 - የተበጠሰው የእንስራው ጆሮ ጠላ ተሹሞበት የነበርው ነው Holonomy of እንስራ/yetebet'esewi ye'inisirawi joro t'ela teshumobeti yeneberiwi newi Holonomy of inisira/(whole) is ጆሮ(part)

Since the relationship between meronomy and holonomy is a whole-part relationship. We could add more examples to clarify the relationship.

Meronomy2- አልማዝ ከሰውነት ከፍሎቿ ጀርባዋን ሸቀሸቃት - የአልማዝ ሙሉ አካል(whole) is the meronomy of ጀርባ/ālimazi kesewineti kefilochwa jeribawani shek'eshek'ati - ye'ālimazi mulu ākali(whole) is the meronomy of jeriba/(part)

Holonomy2- ጀርባዋን የሸቀሸቃት አልማዝ ከአካል ክፍሎቿ መካከል አንዱ ነው/jeribawani yeshek'eshek'ati ālimazi ke'ākali kifilochwa mekakeli ānidu newi/-ጀርባ(part) is the holonomy of አልማዝ(whole)

2.4.4 Antonymy Relations

Antonymy is a well-known relationship that is surprisingly difficult to describe. Not-x is often, but not always, the antonym of the word x. Rich and poor, for example, are antonyms, but not being rich does not mean that one is poor; many people consider themselves neither rich nor poor. Antonymy is a lexical relationship between word forms, not a semantic relationship(Miller et al., 1990). There are different types of antonymy lexical relations like gradation, reverse, converse, and taxonomic sisters.

Gradation: Most antonymy discussions differentiate between contradictory and contrary words. Two propositions are said to be contradictory if the validity of one implies the falsity of the other, and they are said to be contrary if only one proposition can be valid but both can be false. such as, "alive” and “died”. It is thought to be a semantic relationship that organizes lexical memory for adjectives or ordered strings of adjectives in WordNet, all pointing to the same attribute noun like hot and cold(Miller et al., 1990).

Markedness: Most attributes have an orientation. It is natural to think of them as dimensions in hyperspace, where one end of each dimension is anchored at the point of origin of the space. The point of origin is the expected or default value; deviation from it merits comment and is called the marked value of the attribute like right and left.

Converses: These are almost paraphrases that change depending on the point of view. The library is above the store, and the shop is below the library, just as it is above/below, own/belong, and employer/employee.

Taxonomic sisters: This is where the concept of reciprocal exclusivity comes into play. Red and blue belong to the same color family, and anything that is red cannot be blue; they are mutually exclusive.

Antonymy is an asymmetrical relationship, unlike hyponymy and holonomy. In a pattern like neither big nor small, or neither small nor big X and Y words can be swapped out for each other.

Examples of antonymy relation of ambiguous words in the dataset sentences.

Antonymy1 - ሳለ ለሚለው ቃል ተቃራኒው ዶሎዶመ ነው ambiguous word ሳለ/ have three synsets and gloss of the word its opposite in one of the contexts ዶሎዶመ

Antonymy2 - ቀለመ ለሚለው ቃል ተቃራኒው ለቀቀ ነው ambiguous word ቀለመ/ have two synsets and gloss of the word and its opposite in one of the contexts ለቀቀ.

2.4.5 Homonymy Relations

Word class flexibility has often been analyzed in terms of homonymy and polysemy(B. Li et al., 2020). Homonymy is a relationship that exists between lexemes that have the same word form but are not semantically related. Ring ‘a small circular band' and ring ‘make a simple resonant or vibrating tone' are examples of homonymy that vary in word class. A relationship between different meanings of a single lexeme is known as polysemy. In the sense that nominal and verbal use of versatile lexical objects are semantically related, one might argue that word class versatility is analogous to polysemy and should be distinguished from homonymy.

Since homonymy and polysemy occur on a spectrum in practice, it's difficult to apply a clear criterion to distinguish them. As a result, apart from word-class versatility, we will not attempt to tease homonymy. Word class versatility in morphology excludes pairs of lexical items linked by overt derivational affixes, such as to act/an actor. Word class alternations may be due to the existence of a derivational affix in such cases and are thus part of normal morphology. We enable tokens of versatile lexical items to differ in inflectional morphology, on the other hand.

Examples of Homonymy relations of the ambiguous word in the dataset sentences

Homonymy1 - ሳንቃ ማለት ለመዝጊያ ለወለል ወይም ለሌላ የእንጨት ስራ እንዲሆን ከትልቅ ዛፍ በስሱ እየተላገና እየተጠረበ የሚወጣ እንጨት ማለትም ነው።/sanik’a maleti lemezigīya leweleli weyimi lelēla ye’inich’eti sira inidīhoni ketilik’i zafi besisu iyetelagena iyetet’erebe yemīwet’a inich’eti maletimi newi/

From the ambiguous word ሳንቃ its first meaning was ሴቶች ለጌጥና ለዝና ሲባል አንገታቸው ላይ የሚያደርጉት የጌጣጌጦች ጥርቅም ሳንቃ ድሪ ይባላል።/sētochi legēt’ina lezina sībali ānigetachewi layi yemīyaderiguti yegēt’agēt’ochi t’irik’imi sanik’a dirī yibalali/

Homonymy means the word has the same structure with the same sound but in a different meaning.

2.4.6 Troponomy Relations

Synset in the WordNet has various kinds of lexical-semantic relations connected like meronymy and holonomy(between part and wholes), hypernymy, and hyponymy(links between General and specific synsets) in both directions(Lo et al., 2008). Troponymy is a particular kind of entailment that involves temporal co-extensiveness for the two verbs. For example, the troponomy of think is a reason or the troponomy of snore is sleep. Troponymy is a highly polysemous relation whose semantics are domain-dependent, and how troponyms are encoded is not known. The medium distinguishes large classes of verb hierarchies (speak, write, gesture) for communication verbs; motion verbs are distinguished by factors such as tempo or walk vs. run vs. amble(Fellbaum, 2010).

Verb entailment(Fellbaum, 2010) is a fixed truth connection between verbs in which entailment is determined by a portion of the lexical meaning. The entailed meaning is embedded in the entailed meaning in some way. The verb entailment denotes a logical inference relationship: “if x is real, then y must be true” (x ⇒ y). Take the snore-sleep and buy-pay sets. We may assume snore ⇒ sleep and buy ⇒ pay because if someone is snoring, they must be sleeping, and if they want to buy anything, they must pay for it; however, we cannot infer the opposite because one may not snore while sleeping, and one can pay for nothing (not buying, such as donation).

Based on (Ma M., 2003)troponomy is a semantic relation in verb entailment(Fellbaum, 2010) that usually holds between manner elaboration verbs and their corresponding base verbs, if one verb elaborates the manner of another base verb, the two verbs have the troponymy relation.

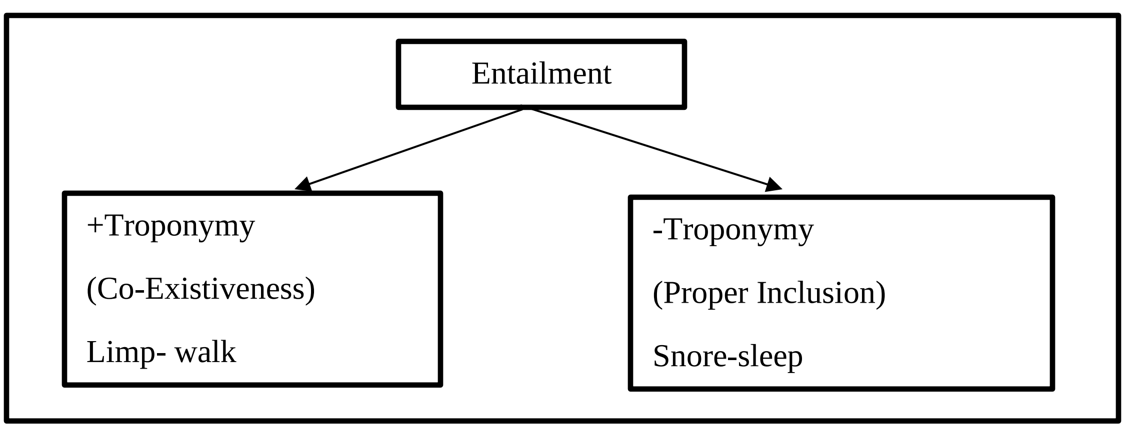


Figure 2:9 Two kinds of entailment with temporal inclusion[(Miller et al., 1990)]

Examples of troponomy relations of an ambiguous word in our dataset

Troponomy1- የተገፋው ሳንቃ በር ተቀረቀረ።/yetegefawi sanik'a beri tek'erek'ere / ገፋ/gefa/ causes ቀረቀረ/k'erek'ere/ exists at the same time and a negative troponomy

Troponomy2- የተለመለመው ቅጠል መሬት ላይ ወደቀ።/yetelemelemewi k'it'eli merēti layi wedek'e/ ለመለመ/lemeleme/ causes ወደቀ/wedek'e/ exists at the same time

2.4.7 Summary

In this section, we would revise the works related to our methodology and further kinds of literature that describe the problem statement to make it clear. Since the main aim of literature review was to show the gap and the methodology to solve the problem. Firstly, we introduced the Amharic language and its structure of morphology within its grammatical structure because of the research done on the language. Secondly, we define briefly different language models to identify which model is comfortable to solve the our problem. The language models that have the potential to solve text classification models including contextual embeddings like BERT, ELMO, GPT presented. And then we discussed the properties of text processing language models RNN family, especially the properties of CNN, LSTM, and Bi-LSTM within activation function and optimizer. Thirdly, different approaches for solving word sense disambiguation problems like knowledge-based, corpus-based and semi-supervised approaches. Lastly, we described Amharic WordNet and its relations like synonymy, hypernymy/hyponymy, meronym/holonomy, homonomy, Antonymy, and Troponomy with specific examples.

2.5 Related Work

2.5.1 BERT for word sense Disambiguation

Based on (Huang et al., 2019)word sense disambiguation is a basic task and a long-standing problem in Natural Language Processing, to determine the precise meaning of an ambiguous word in a given context. In a supervised neural WSD method, this study makes better use of gloss knowledge. Pre-trained language models, such as ELMo and BERT, have recently demonstrated their effectiveness in reducing feature engineering efforts. BERT has excelled in question answering and natural language inference, in particular. They create context-gloss pairs from the target word's glosses in all possible senses in WordNet, treating the WSD task as a sentence-pair classification problem. To solve this dilemma, the authors used the SemCor3.0 training corpus to fine-tune the pre-trained BERT model.

Rather than the previous works they exploit the semantic relationships between senses such as synonymy, hypernymy, and hyponymy and rely on pre-trained BERT word vectors (feature-based approach); they leverage gloss knowledge or sense definition and use BERT through fine-tuning procedures. The authors used a pre-trained uncased BERTBASE model for fine-tuning, rather than the BERTLARGE model in their task. The models bullied by the parameters of Transformer blocks are 12, the number of the hidden layer is 768, the number of self-attention heads is 12, and the total number of parameters of the pre-trained model is 110M. When the time fine-tuning, to find the best settings for our studies, they used the development set (SE07). They keep the chance of falling out at 0.1 and the number of epochs at four. The batch size is 64 and the initial learning rate is 2e-5. Finally, by this hyperparameter experiment, they got a state-of-the-art result.

2.5.2 WSD Using Pre-Trained Contextualized Word Representations

Based on the study (Hadiwinoto et al., 2019)identifying the correct word sense in a given context or assign a pre-defined sense to a word known as word sense disambiguation. The WSD method assigns a meaning to a word based on its context, which includes the other terms in the sentence. The WSD scheme assigns a meaning to a word by considering its context with the other terms in the sentence. The pre-trained model used a BERT bidirectional transformer encoder model that masked words and predicted the next sentence.

Those used BERT model for WSD leveraging nearest neighbor matching, and linear projection of hidden layers including simple last layer projection, layer weighting (LW), and gated linear unit. In their linear projection trained the model by Adam optimizer, learning rate $10^{-3,}$ and the parameter updated per mini-batch of 16 sentences. Based on their experiment BERT, a pre-trained contextualized word representation that captures the context in the hidden vector. As a result, the linear projection of the hidden vectors, coupled with gating to filter the values, given the prior state of the art.

2.5.3 WSD for Punjabi language

(P. Kumar, 2020)attempted word sense disambiguation system for the Punjabi language by using deep learning techniques. According to the authors, a word sense disambiguation scheme identifies the correct meaning of a word based on its context. In their research, they used multilayer perceptron and LSTM methods on a 66-word dataset. Six traditional supervised machine learning techniques were also tested on the same dataset using unigram and bigram feature sets as part of their research.

However, Deep learning is a multilayer Neural networks approach that learns directly from a corpus and thus eliminates the need for a manual feature extraction process. While the performance of traditional machine learning techniques depends on the efficiency of the feature extraction process and the quality of annotated text datasets, deep learning techniques use word vectors as a feature that captures very complex characteristics of data without much manual selection of features. Based on their study result multilayer perceptron outperforms traditional machine learning algorithms and LSTM due to its more than one hidden layer.

2.5.4 WSD for Chinese Language

Meng (2020) attempted graph-based Chinese WSD method based on English word similarity computation with mapping the words in ambiguous sentences and then calculates the similarity of English words to further build a disambiguation graph. Finally, the graph scoring algorithm is used to score the importance of word sense vertices to obtain the final correct word meaning which has a higher score. The step-by-step process used in the graph and word similarity for word sense disambiguation follows:

Firstly, the content words in ambiguous sentences are processed by word meaning mapping to obtain the English word set corresponding to the context. Secondly, the similarity calculation method based on word vector and knowledge base is used to compute the similarity of the obtained English word set. Thirdly, using English words as vertices, semantic relations between words as edges, and similarity values as edge weights, a disambiguation graph is constructed. Fourthly, the graph scoring algorithm is used to score the importance of the word vertices in the disambiguation graph, and the meaning score of each word to be disambiguated represented by the English word is obtained. Lastly, according to the scores of each word meaning obtained by the scoring of the graph, the one with the highest score is selected as the final correct word meaning.

The experimental dataset comes from the international semantic evaluation task SemEval-2007 with the multilingual Chinese English lexical sample task. It contains 19 nouns and 21 verbs, a total of 40 ambiguous words. Using the official standard evaluation method, the macro average was maximum. Generally, the study used effectively use English knowledge resources and obtain a disambiguation accuracy rate of 0.451, which can improve the accuracy of Chinese WSD.

2.5.5 WSD for Hindi Language

Sharma & Bhatia, (2008) WSD model was created for the Hindi language. The author employs a variety of approaches to word sense disambiguation (WSD), including knowledge-based, machine learning-based, and hybrid approaches. The problem of word sense disambiguation is being attempted to be solved in this study using the Hindi WordNet built at IIT, which contains various terms and their sets of synonyms known as synsets. By comparing the various meanings of words in the sentence with the word present in the synset form of WordNet and the details related to these words in the form of parts-of-speech, the researcher attempted to overcome the ambiguity. Finally, the researcher discovered that knowledge-based methods are the best method for word sense disambiguation, and Lesk's algorithm was used as an example to demonstrate its applicability for WSD in Hindi. The author took an example paragraph and produced its meaning bag, then extracted the semantic bag for the word to be disambiguated, and then did the overlap of both bags corresponding to each sense of the word, and finally found the right sense of the word. However, the researcher does not use a knowledge-based approach to evaluate the output of the developed Hindi WSD method.

The other WSD for Hindi has been introduced by (Bala, 2013) With Hindi WordNet, the researcher created a WSD tool using a knowledge-based approach. WSD was investigated in terms of the field in which certain words in meaning are sense-tagged. There are two forms of WSD tasks: first, tag all significant words (nouns, verbs, adjectives, and adverbs), and second, tag some significant words (usually nouns or verbs). In this study, parts-of-speech (POS) and Hindi WordNet were used to enhance the method. If a word appears several times in the same text in various senses, the algorithm is most likely to assign the same synset or synonyms to all of its occurrences. To test word sense disambiguation, the researcher used a 43-word corpus of collocations and occurrences. The researcher had a precision of 58% and a recall of 50% achieved.

2.5.6 WSD for Arabic Language

Bouhriz et al., (2016) Word Sense Disambiguation consists of identifying the correct meaning of an ambiguous term occurring in a provided meaning. Most Arabic WSD systems were based on the details derived from the local sense of the term. For the best disambiguation, this detail is normally insufficient. To address this problem, they proposed an Arabic WSD scheme that is based not only on the local context but also on the global context extracted from the entire text. The aim is to merge local contextual information with global contextual information to improve disambiguation. The proposed model selects word senses from the Arabic WordNet (AWN) resource. The meaning described to an ambiguous term is therefore the one that is nearest in semantic proximity to both the local and global contexts. The semantic hierarchy provided by WordNet is used to calculate this proximity. The study's general architecture describes the following internal structure.

- Sense inventory: consists of selecting the senses of the words.

- Context representation: represents senses and contexts in a formal manner.
- Disambiguation Process: attributes for every ambiguous word its correct sense according to its context.

The sense inventory phase is the one that distinguishes one method from another, depending on the methodology used. The Knowledge-based approach, like based on the use of external lexical tools. These tools include all of a language's vocabulary as well as its senses. Dictionaries, thesaurus(Yarowsky, 1992), and ontologies are examples of these tools. The second approach, unlike the first, does not rely on external lexical tools, instead of obtaining the requisite information to define words' senses from a corpus; it is known as a Corpus-based approach. The application of statistical language models to this corpus yielded this knowledge.

In this step, a preprocessing phase is applied; it contains a text segmentation process, a stop words removal process, and finally a stemming process to remove words' affixes (prefixes and suffixes). Afterward, the obtained words are classified, according to the AWN, into two categories, non-ambiguous words belonging to one synset, possessing one sense, and ambiguous words belonging to several synsets possessing several senses. In the second step of context, representation consists of representing words' sense as a vector. The final step is to assign the acceptable meaning to each ambiguous word. This is accomplished by selecting the meaning that is most semantically related to the local and global context. The percentage of vectors in this context that is close to the vector of this meaning defines sense semantic proximity with this context. For each ambiguous word sense, the local and global semantic proximity is determined, and a pair of percentages reflecting each of the semantic proximity is obtained. Finally, the ambiguous term will be assigned to the meaning with the stronger average of its two percentages. The proposed model, unlike other models, takes into consideration two types of context during disambiguation procedure The first is the local context, which is determined by words. First in the local context of the ambiguous word, and secondly the overall context established by the entire text. The experimental result achieved 74% accuracy.

2.5.7 WSD for Amharic language

Word sense disambiguation on the Amharic language done by different methods in different scholars. However, no one correctly disambiguates by using machine learning approaches. Some of them were analyzed as follows.

M. Solomon (2010)employed a supervised machine-learning approach for Amharic WSD. The supervised method of word sense disambiguation used a machine learning approach to learn and classify from labeled training data, that encoded in terms of several features together with their appropriate sense label. They used manually annotated training data containing instances of target words to learn the context in which target words are used.

Machine learning algorithms make use of the instance attributes or features in the training data and generate a model to predict the label of any given instance. The applied model can be applied to unseen instances to predict their labels. Algorithms that can learn to predict discrete-valued labels are called classification algorithms or classifiers, whereas the algorithms that can learn to predict continuous valued labels are called regression algorithms.

M. Solomon (2010) For Amharic word sense disambiguation, a supervised machine learning method using the Naive Bayes algorithm on the Weka 3.62 kit, the classifier algorithm was used to classify a word into the correct meaning. The dataset of this study was five ambiguous words from British National Corpus (BNC) within 1048 senses and translate back to Amharic in monolingual corpora of the English language. The challenges of this study were a problem of knowledge acquisition bottleneck, where the amount of labeled data provided to the classifiers is limited. They used 100 Amharic sentences for training and testing and got 73% up to 83% accuracy result.

A. Solomon (2011)further employed Amharic word sense disambiguation by using an unsupervised machine learning approach. This study aimed to find a solution to the problem of automatically determining the correct meaning of an ambiguous word based on its context. Word sense induction or discrimination techniques are used in unsupervised methods to automatically discover senses from unlabeled corpora and then add them to WSD.

They used unannotated training data containing instances of the target word during their study. The dataset used in this study were አጠና (eTena), መሳል (mesal), መሣሣት (me`sa`sat), መጥራት (metrat), and ቀረጸ (qereSe) Amharic Ambiguous words. The sense examples were translated to Amharic using the Amharic-English dictionary and preprocessed to make it ready for experimentation from British National Corpus (BNC). And then it translates to Amharic by using a bilingual dictionary. The challenges of this research were a lack of Amharic resources like that of other researchers. To overcome the lack of data it is recommended to test bootstrapping approach which requires little training data. And it is recommended to increase the number of ambiguous words covered and test other approaches such as knowledge-based and hybrid. But those researches done on only five ambiguous words and in a specific verb class are the great limitations of research.

(Wassie et al., 2014)also researched Amharic WSD by using a semi-supervised approach. They applied both unsupervised and supervised algorithms for clustering based on instances similarity and classification after the unlabeled data respectively. The semi-supervised approach train initially a classifier with a small-sized tagged corpus and then brings further improvements in the process of learning. Those used different algorithms like Bootstrapping and decision tree within 10-10 window size in 481 sentences within 5 ambiguous words dataset. This approach nearest to supervised, and whose aim is to learn a model with rare annotated and several unannotated data. However, annotated data are costive even limit the weight-related to unsupervised.

Yesuf & Assabie(2017) attempted Amharic word sense disambiguation by knowledge-based approach for Amharic all words task by using WordNet containing thousands of words with their related sense. They implemented overlap-based algorithms and preprocessing tasks using python and Java programming. During their two experiments, the first one to evaluate the effect of WordNet with or without morphological analysis but WordNet with morphological analysis had a good performance. The second experiment was done due to determining the optimal window size and the two-word window founded to be the optimal window. The model expected to disambiguate all open class words in running but only one word disambiguates at a time that is the main limitation. Like that of other scholar's lack of Amharic linguistic resource like WordNet were the main challenges of this study.

Tesema (2016)attempted WSD by hybrid unsupervised and rule-based approaches for 20 most frequent Oromo words having a corpus containing thousands of Oromo sentences. The authors used to partition and hierarchical clustering algorithms and manually crafted rules. Link algorithms and Agglomerative from hierarchical and k-means expectation-maximization from partition clustering were implemented using Weka 3.75. during their experiment, the optimal window size was two and achieved 89.47% accuracy rather than supervised since 76.05%. And during their experiment partition clustering was better than hierarchical clustering. Manually developed rules are prone to error, time taking and difficult to cover many numbers of words were the main limitation of this study.

The research (Dureti, 2017; Yesuf & Assabie, 2017)attempted WSD system by using WordNet. The ambiguous word and its senses are located in a database and used as a knowledge base to disambiguate a given word. The WordNet hierarchy contains a maximum of three senses for a word is implemented on the Lesk algorithm. The actual disambiguation process, during the experiment, was only on the use of gloss of the word itself, not related words. The system expected to disambiguate all open class words in a given sentence such as nouns, adjectives, adverbs, and verbs. Nevertheless, it disambiguates only one frequently occurring target word in the WordNet for an input sentence. In addition to that, the lesk algorithm includes multiple words having multiple senses were considered at once problem during executing of the system.

Mieraf (2020)attempted Amharic hierarchical word sense disambiguation system by using WordNet at sentence level in all classes of a word. They included a verb, noun, adverb, and an adjective class of the word in their research and in the sentence, level differs from the previous works in the area. To disambiguate Amharic sentences used context-to-gloss overlap and augmented semantic space approaches. The most popular algorithm extended lesk (Banerjee & Pedersen, 2002)used for word sense disambiguation for Amharic language. This research started to solve the problem of Lesk algorithm when multiple words having multiple senses are considered at once (combinatorial explosion problem).

One of the methods applied in this research was augmented semantic space is highly dependent on the number of words that exist in the WordNet since 17 words are only included in the WordNet. It also depends on several context words in the sentences which also exist in the wordnet. The algorithm works counting overlap between glosses which makes it dependent on the length of glosses, the exact wording of the number of related synsets and glosses. The context-to-gloss overlap is limited to perform well for short sentences. Short sentences have a smaller number of context words which makes the frequency count smaller were the main limitation of the approach. The WordNet didn't consider antonyms, homonymy, troponym, and holonomy relation of the words have single sense up to three senses were the main limitation of this study.

The challenge for this research was getting test sentences for the experiments because there is no sense-tagged Amharic corpus prepared for WSD or other applications. In other languages like English, there are many senses tagged corpora like SemCor which are open to being used in different researches. And after all the researcher recommends the augmented semantic space can be improved if all relationships between synsets are considered and more single sense words are added to the WordNet. The context-to-gloss can be improved by increasing the number of related synsets in the WordNet.

Mulat (2020)also automatically develop WordNet from a large corpus by using word embedding (CBOW and skip-gram) methods. The WordNet generates hypernym, hyponym, synonym, and antonym relations of a word from a large corpus. To generate synonyms, they used cosine similarity, and for hyponymy, hypernymy extraction mutual information concept and measuring similarity methods used respectively. Lastly, antonym relation from the trained model by the concept of analogy and achieved a good result in wordnet extraction. However, they didn't include the meronym, holonomy, and troponymy relation of the word was the main limitation.

2.5.8 Summary

Related works were reviewed to show the state-of-the-art and to make the problem clear. Instead, we reviewed different word sense disambiguation in Amharic language up to recent last work. During this review, most researchers face the challenges of resource limitation constraints like that of other language WordNet, SemCor, and sentences datasets limited in Amharic. The researcher (Mieraf, 2020) included all classes of the word rather than the previous researchers since they were done on verb class only senses in the WordNet. However, this research lacks antonyms, homonymy, troponym, and holonomy relation in their synsets. This gap solved by this research. On the other hand, there are a lot of researches done on WSD in other languages as stated some of in above like Chinese, Punjabi, and others for clear understanding of algorithms like WSD using pre-trained contextualized word representations even if resource limitation to implement these methods.

Chapter Three

3 Design of Automatic Amharic Word Sense Disambiguation Model

3.1 Introduction

In this chapter, we are discussing the design part of the proposed automatic Amharic word sense disambiguation model using a deep learning approach. Besides, we deal with different text pre-processing tasks like tokenization, stop-word removal, normalization, and morphological analysis. And also, we discuss the training activity of the deep learning model and evaluate the performance of the model by using the dataset.

3.2 Model Architecture

As presented in figure 3:1 below describes the general model of automatic Amharic word sense disambiguation model starts from collecting Amharic corpus from different sources. Firstly, finding 159 ambiguous word which has multiple meaning in a different context. And then defining those words based on their variant synset or gloss within word relations from Amharic dictionary and sketch engines. Then building a lexical database that has a relationship of ambiguous words, their synset, gloss, and relations. After building our WordNet preparing a dataset sentence based on the relation and reviewed by expertise. The dataset sentence was given for tokenizer algorithm and passed the listed preprocessing technique up to stemming, padding text to sequence. If the data is ready for training our work gone to building our deep learning model within a specified hyperparameter experimental setup.

Then compiled our data with our specified model, and trained our model by fitted data in different rounds or epochs starting from default parameters and changed experimental setup until gotten minimum loss and higher accuracy. Validating and testing the performance of the model would be the last task of this thesis work. Since, the model had three models preprocessing module, model development modules, and feature mapping or string-matching trained model with WordNet. In the last module if the output of the SoftMax classifier results in prediction less than 8 levels then fetching its gloss from the wordnet in order to more clarify the ambiguous words meaning to the user. Therefore, our model performance measured by precision, recall and F1 score by calculating confusion matrix within the testing dataset and disambiguating ambiguous words in a predicated sentence would be the output of our model.

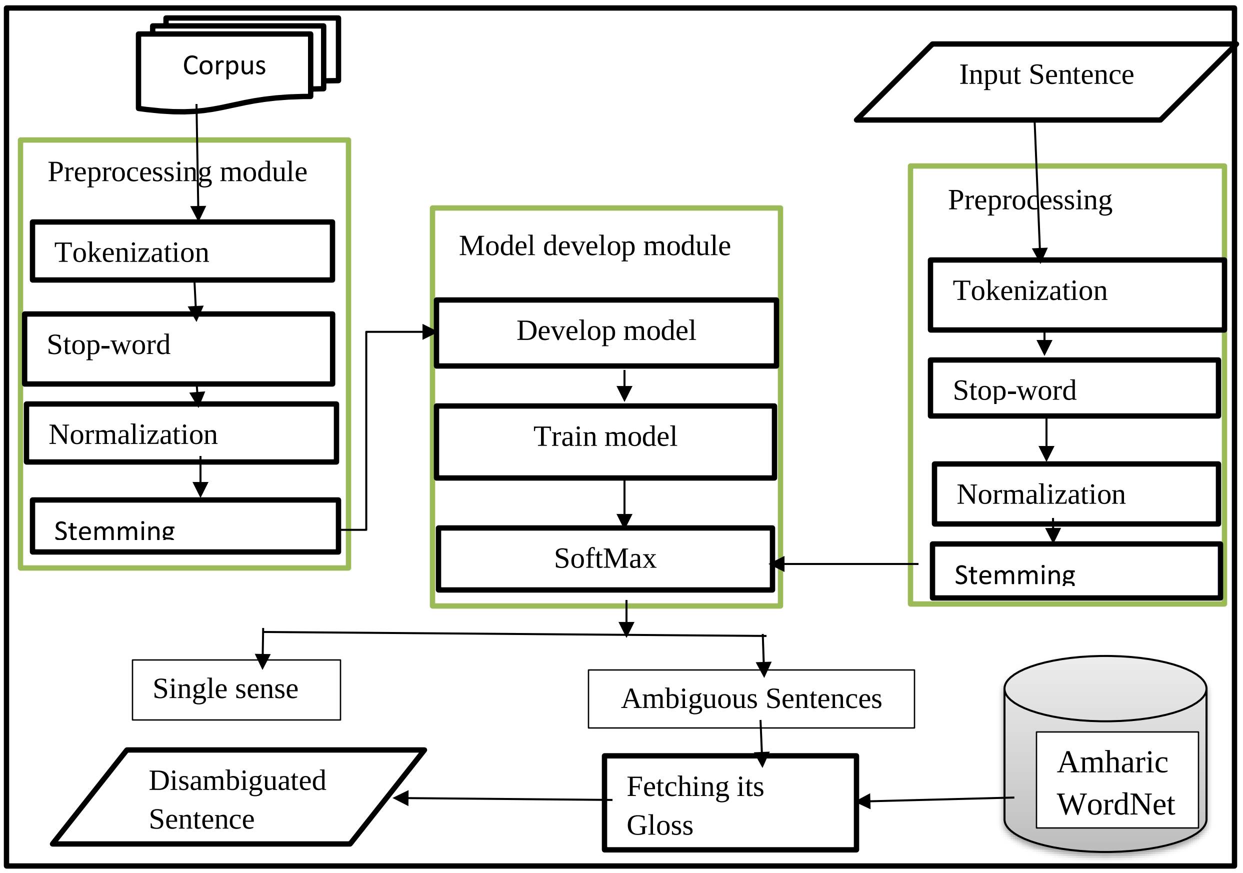


Figure 3:1 Proposed General flow of Automatic Amharic Word Sense Disambiguation model

3.3 Preprocessing

In this section, the raw text converted into meaningful data using text preprocessing. Preprocessing is to remove raw data that contain unwanted punctuation marks, stop-words, numerical value, and special characters, and replace with a single alphabet (normalization) in a different representation. Because of reducing the effect on the performance and the correctness of the model.

3.3.1 Tokenization

Splitting data into a small chunk of words or the whole corpus splitting into word level known as tokenization. it takes the input text supplied from a user or in the given corpus and split it into a sequence of tokens, which is the process of breaking a stream of text down into words. And finally, it gives the list of words that are used in the next phase of preprocessing. The model that we use may also split into character levels because Amharic special characters are not recognized as that of other languages.

In most Latin language white spaces and other punctuation marks like question mark {?} are used as the main approximation of word-to-word delimiter (boundary markers between sequences of words). Like the other languages, the Amharic language also has its punctuation marks which separate texts or sentences into a stream of words. Amharic punctuation marks include 'ሁለት ነጥብ'(፡)/two points/ ኮለን(,) /colon/ 'አራት ነጥብ(።)/four-point/, 'ነጠላ ሰረዝ'(፣)/single comma/, 'ድርብ ሰረዝ' (፤)/double comma/, ጥያቄ ምልክት?'/question mark/, ቃል አጋኖ (!)/Word exaggeration/,ይዘት(.)/dot/, and ትምህርተ ስላቅ(¡)/Education sarcasm/are used as sentence delimiter or as white space. In this thesis work, we remove all punctuation marks even if the tokenizer tokenizes as a word but its handle by using word remove the punctuation at index -1 if the word has punctuation marks.

3.3.2 Normalization

Tokenization segments or split the given text into words. In this study during normalization converts a list of words to a more uniform sequence since, in the Amharic language, three types of normalization issue arise. The first one is compound words think as a different word in the system like 'አዲስ' and 'አበባ' but this is a single word in Amharic. We handle this type of words by a list of double words and substitute like አዲስ_አበባ if consequently occur. The second type of normalization which contains a slash (/), dot(.) occurs sometimes as a title like ት/ቤት expands and write as ትምህርት_ቤት. Lastly, the Amharic normalization made due to consistent writing styles like አላማ ዐላማ ዓላማ substitute {አ፣ ዐ፣ዓ} by አ.

Algorithm 1: Normalization of Amharic characters

```
Open normalization. text file
Read each word w in the input text
nor =read each line in normalization.txt
for i in w:
for j in nor:
x=j.split( )
if iequals to_(x[0]):
Replace i by x[1]
```

3.3.3 Stop Word Removal

Stop words are commonly eliminated from many NLP applications because these words disturbing the meaning of important terms in a document since it covers 80% of the document it needs additional memory and computational time during the training of the model. There are no common or universal stop words in Amharic because of context-dependent like that of other languages. The most common words appearing with high frequency in every text in a language and also, they are low information-bearing words. In the Amharic language, most of the time, conjunctions, articles, and prepositions occurred frequently and they don't give any useful information in the text. In our task, we are predicting the context of words from a word surrounding it using word embedding. So, the reason behind the elimination of stop words from our corpus before training Bi-LSTM model is to improve the power of predicting the context, to reduce memory overhead, get a better result since we are focusing on the important terms it reduces noise and false positive. Therefore, the word like በዚህ ፣ እንደ ፣ ነገር ፣ እና ፣ ና ፣ ወይም … occurred in the corpus remove by a list of stop words.

Algorithm 2: Stop-Word Removal

```
Input: Amharic text corpus
1: start
2: open Amharic corpus from our disk
3: read word tokenizers input from a local machine
4:      Apply tokenizer
5: Read Amharic alphabets from disk
6:      go to line 3
7:      check characters
8:       if step 7 is true:
9             replace characters
10:                else:
11:                    jump
12: Read compound words from the disk
13: Read titles from disk
14:             go to line 3
15:                if true:
16:                     combine it
17:                else:
18:                    jump
19: Read stop words from the disk
20:            go to line 3
21:            go to line 5
22:            go to line 12
23:            go to line 13
24:                if true:
25:                     remove it
26:                else:
27:                    jump
28:     else:
29          end of file
30:      write the file into the disk
31:       end
Output: Clean Data
```

3.3.4 Stemming

Stemming is the process of removing additional suffix, prefix, and infix of a word and remain the stem word for every word in sentences. For a morphologically rich language like Amharic, in the time of stemming removing affixes. it is an important preprocessing step in Natural Language Preprocessing applications.

Mainly the purpose of stemming in the area where two or more words that have the same root or stem have given the same vector representation during word embedding. For this reason, stemming were the parts of our preprocessing step in this study also even if our stemmer has its limitation. For example, the words (ሰባሪዎች፣ ተሰበረ፣ ሊሰበሩ፣ እንሰብራቸዋለን፣ ተሰበሩ፣ ተሰበረች፣ ልትሰበር) or as a future (ይሰብራል፣ ትሰብራለች፣ ይሰብራሉ፣ እሰብራለን) or as a past (ሰብራለች፣ ሰብሯል፣ ሰብረናል፣ሰብረዋል) have root ስ-ብ-ር it's better to save time and memory during training also the other advantages of stemming. Analogously in English the word Break has present =Break, past =Broke and participle = Broken for singular we can add 's' and for plural, we could use as it is like 'break, broke, broken' rather than breaks in singular format. This was the reason for how much Amharic language morphologically rich rather than English. In this study special rules removing infixes and under and over stemming was the problem of our algorithm. Such problems are solved by the manual stemming technique at the end of the preprocessing.

Algorithm 3: Stemming Algorithm

Input: Normalized Amharic corpus
1: start *2: Open suffix.txt and prefix.txt* *3: For word in input text* *4: If a word starts with prefix* *5: Remove prefix* *6: If a word ends with suffix & not in the exception list* *7: Remove suffix* *8: Return stemmed normalized corpus*
Output: Normalized and stemmed Amharic corpus

3.4 Amharic WordNet

WordNet for Amharic language has various advantages like machine translation, question answering, event extraction, and so on. In Amharic WordNet the words which had similar meaning and synset grouped into one. Amharic WordNet is a system that connects various lexical and semantic relationships among Amharic words. The Amharic WordNet was created using the same principles as the English WordNet manually even if time taker its effective approach. In this research, Amharic WordNet helped to fetch its gloss and disambiguate an ambiguous word for the user since it is a useful tool for computational linguistic and Natural Language Preprocessing.

3.4.1 Structure of Amharic WordNet

Our Amharic WordNet structure builds the words, their synset, gloss, and relation of an ambiguous word. Synsets are basic building blocks of WordNets for disambiguating ambiguous words in a different context. For example in our WordNet ሳለ/ sale/ have four different senses.

Synset 1(ሳለ)፡ ሞረደ፣ አሾለ፣ አተባ፣ ስለት አወጣ/morede፣ āshole፣ āteba፣ sileti āwet’a/

Synset 2(ሳለ)፡ በጉንፋን ወይም በኮሮና ምክንያት ኡህ ኡህ አለ ወይም ተነተነ/begunifani weyimi bekorona mikiniyati uhi uhi āle weyimi tenetene/

Synset 3(ሳለ)፡ የአንድን ነገር ምስልና ቅርጽ በማስመሰል ወረቀት ላይ በእርሳስ ሳለ/ye’ānidini negeri misilina k’irits’i bemasimeseli werek’eti layi be’irisasi sale/

Synset 4(ሳለ)፡ ከቦታው ሳይንቀሳቀስ እያለ እንደማለትም ነው/kebotawi sayinik’esak’esi iyale inidemaletimi newi/

The four synset in the WordNet have different synset ID, gloss ID but the same word ID. Therefore, this ambiguous word have 4 sysnset 4 glosss and 8 relations.

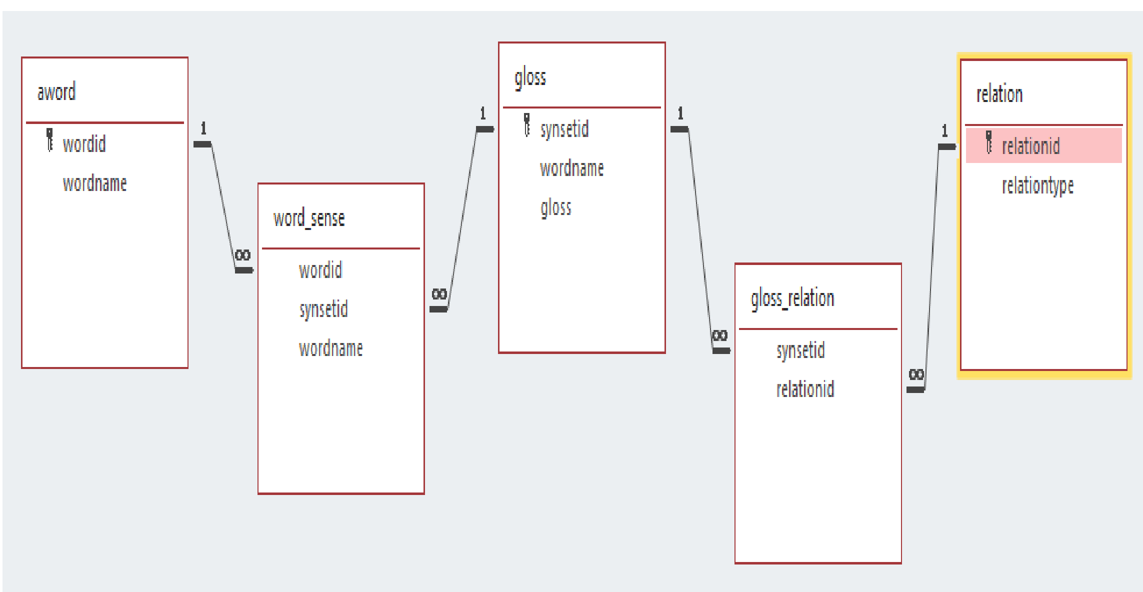


Figure 3:2 Amharic WordNet database schema

As shown in figure 3:2 above our Amharic WordNet have 5 tables such as ambiguous word, gloss, relation, word-sense, and gloss relation. The relationship between ambiguous word(aword) and gloss is many-to-many like that of gloss and relation.

3.5 Model Development

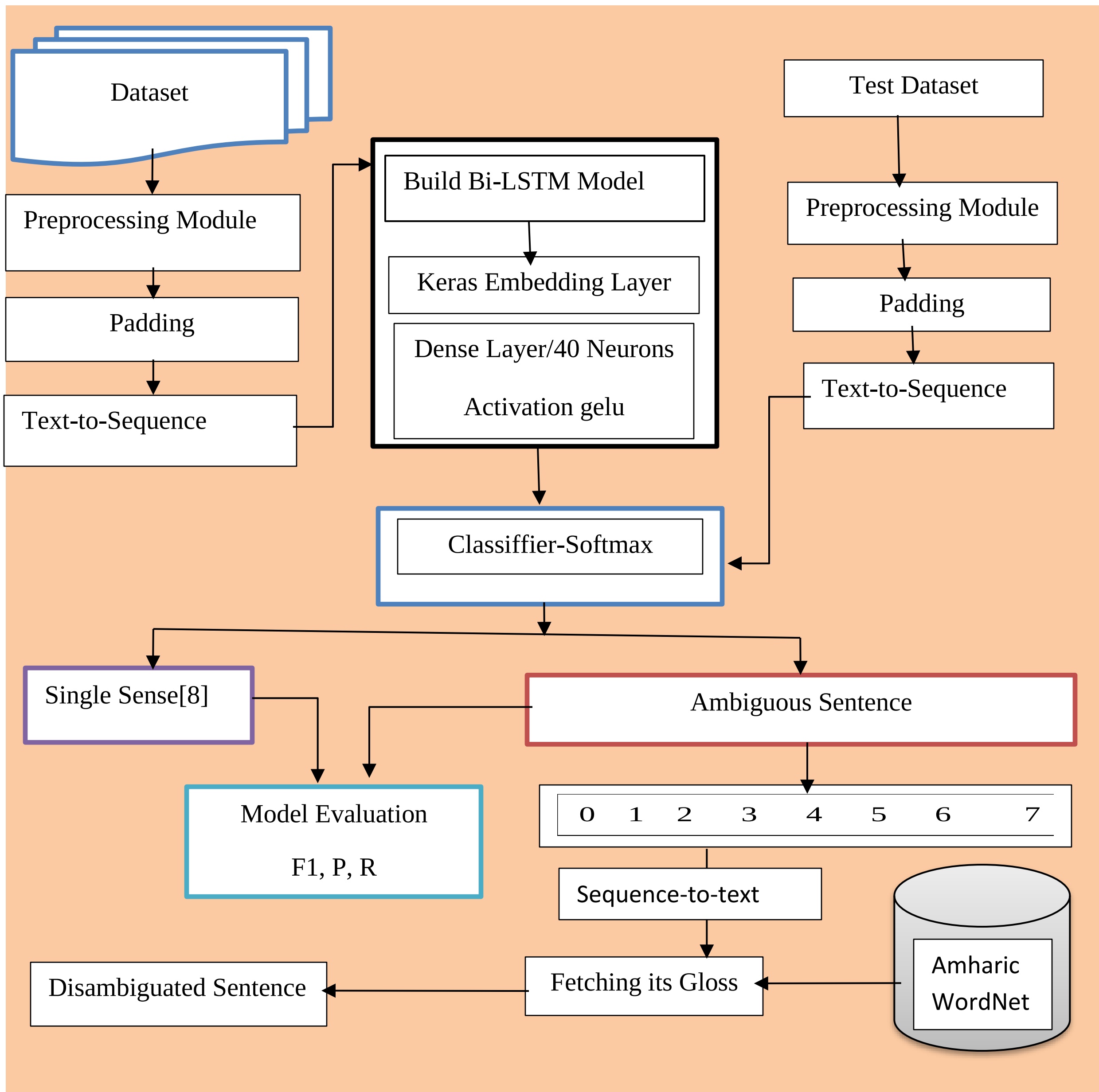


Figure 3:3 Proposed Bi-LSTM Model Architecture for Amharic WSD

After preprocessing step, designed three text processing language models within specifying hyperparameters. From that hyperparameter (vocabulary size, embedding dimensions, input length, Dropout rate, the number of neurons, and activation function) determine in the model development stage. For the word sense disambiguation problem by CNN model, the following parameters were set. Vocabulary size, model regularize, the number of Global Max-Polling layer, Convolutional layer, Dropout rate, Number of dense layers, and activation function specified.

On the other hand, for both model's LSTM, Bi-LSTM the same hyperparameters were used to build vocabulary size, embedding dimensions, input length, Dense layer, and activation function. During the model training stage in this study, all specified hyperparameter compile with their model metrics and fit the data split before for training, testing, and validation. In training specifying parameters were the number of epochs/ learning round, batch size, learning rate, data shuffle size. After the model training the last phase evaluating the performance of the model based on its training and testing data. We compare and contrast all three model performances by performance metrics (precision, recall, and f-measure). After validated and tested from the three models the one which has higher accuracy becomes used to solve the specified Amharic word sense disambiguation problem. Bi-LSTM neural network language model selected from the remaining two based on its good performance and suitable for such problems.

Chapter Four

4 Experiments and Evaluation

4.1 Introduction

To validate the proposed Automatic Amharic word sense disambiguation system a series of experiments were done in this chapter. This section of the research discusses also data collection and evaluation of the model.

4.2 Experimentation

4.2.1 Data Collection

The research dataset started with finding ambiguous Amharic words from different sources like Amharic sketch engine, Amharic dictionary, within its glosses or definitions, and then constructing sentences by using their definition for training and testing were the second task of the research. By using ambiguous words within its definitions and relation type prepared Amharic WordNet. And then reviewed by two experts to annotate the dataset, if two agreed then selected as a dataset but if they do not disagree with the researchers' idea consider it during the decision. Single sense-data sentences which do not have ambiguous words collected from different websites. Generally, the dataset and the number of ambiguous words within its synset summarized as follows.

Table 4:1 The Dataset of this Study

Number of Ambiguous Words	Number of Synsets	Number of Gloss	Sentences in all Relations	Source of Data
159	1350	460	2164	Amharic dictionary, sketch engines, Abyssinica, and experts

Table 4:2 The number of sentence dataset for training, testing, and validation

No	Relation type	Training Data
1	Synonym	169
2	Hypernyms	169
3	Hyponymy	169
4	Meronomy	169
5	Holonomy	169
6	Antonymy	167
7	Homonymy	169
8	Troponomy	169
9	Single sense	428
Training Data size		1778
Validation Data size		210
Testing Data size		176
Total sentences in the dataset		2164

4.2.2 Implementation

In this experiment, we use anaconda software tools since its free open-source distribution of programming languages like python which is used for data science, machine learning in the area of artificial intelligence. From Anaconda we used the Tensor Flow environment within Jupiter code editor. Python is an object-oriented programming language that provides rapid application development. It is high-level programming. Python allows you to develop desktop and web applications. Whereas Jupiter is a scientific development environment for python, it includes editing, interactive debugging, testing.

4.2.3 Hyperparameters

The hyperparameters used in this research description as follows:

Loss Function: One of the most essential components of deep learning models is the loss function. The loss is the model's prediction error, which is used to measure the gradients. The Loss function is a method used to calculate loss. During preparation, the error function is used to measure the model's error. The neuron weights in a specific layer are changed using error backpropagation so that the error rate decreases in subsequent evaluations. We used categorical cross entropy in our system, which is used for a multi-class classification task, which is defined as(Maslej Krešňáková et al., 2020).

$$BCE = -\left(ylog(\hat{y})(1-y)\log(1-\hat{y})\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Equation 4-1 Equation of Categorical Cross-Entropy

where y represents the ground truth and $\hat{y}$ represents predicted value

Optimizers

Gradient Descent: Gradient descent is one of the most widely used optimization algorithms and the most common method for optimizing neural networks. Gradient descent is a technique for minimizing a cost function J(θ) with model parameters by updating the parameters in the opposite direction of the cost function ∇θJ(θ) gradients for the parameters. The size of the measures we take to achieve a local minimum is determined by our learning rate. There are three types of gradient descent, each with a different amount of data used to calculate the objective function's gradient. Batch gradient descent, stochastic gradient descent, and mini-batch gradient descent are the three forms of gradient descent(Kim et al., 2017).

Stochastic Gradient Descent: SGD is a variant of Gradient Descent. It tries to update the model's parameters more frequently and it requires less memory as no need to store values of loss functions were its advantage. It has a high variance in model parameters and to get the same convergence as gradient descent needs to slowly reduce the value of learning rate were its disadvantages(Zou et al., 2019).

Adaptive gradient descent (Adagrad):- Adagrad is an algorithm for gradient-based optimization that avoid the concept of using fixed learning rate and uses dynamic learning rates for every parameter. They are tailored to the frequency at which a parameter is updated during training using parameter-specific learning rates. This implies that for infrequent parameters, larger updates are performed and for regular parameters, smaller updates are performed.

daptive delta (Adadelta):- It is another improvements from Adagrad. It is a robust extension of adagrad that adapts learning rate based on a moving window of gradient updates, instead of accumulating all past gradients. Delta in Adadelta refers to the difference between the current weights and newly update weight.

Adam: Zou et al. (2019)optimization function is used to reduce the model's error rate in prediction. In this research, we used Adam (Adaptive Moment Estimation) optimizer. Adam is an optimization function that computes each parameter's learning estimate(Maslej Krešňáková et al., 2020). Like RMSprop, Adam has an exponentially decaying average of past gradients v(t) as well as an exponentially decaying average of past squared gradients m(t). Since all moving averages are set to zero, the calculation of the moments is skewed towards zero. This happens most often in the early stages of decay when the decay parameters ($\beta 1, \beta 2$) are similar to 1. Such biased can be removed using modified estimations $m^{\wedge t}$, $v^{\wedge t}$:

$$m^{\wedge t} = \frac{mt}{1-\beta t1} \ , \ v^{\wedge t} = \frac{vt}{1-\beta t2} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (2)$$

Equation 4-2 Equation of Adams Exponentially decaying of past squared gradient(left) and past gradient(right)

where learning rate α= 0.001 and 10^{-8} for *e* parameter

$$\beta 1 = \text{The exponential decay rate for the first moment estimates}(0.9)$$

$$\beta 2 = \text{ The exponential decay rate for the second} - \text{moment estimates}(0.999)$$

Nesterov-accelerated Adaptive Moment Estimation (Nadam):- It is a combination of Adam and Nesterov. Nadam uses nesterov to update the gradient one step ahead by replacing the previous ˆm of adam optimizer with current ˆm. The learning process is accelerated by summing up the exponential decay of the moving averages for the previous and current gradient.

Even if a lot of optimizers, in this research from the above optimizers adam was effective for solving our problem have a very good learning rate rather than the others.

Dropout: - One of the methods used to reduce overfitting is a dropout. We have applied dropout as a variable hyperparameter during the three experiments but the one comfortable for our models was 0.39. Dropout randomly disconnecting inputs from the first fully connected layer to the second fully connected layer. It enables multiple, redundant nodes to activate when given with a similar pattern which helps our model to generalize and lead to reduce overfitting(Park et al., 2019).

Maslej Krešňáková et al. (2020)dropout is needed during model training, individual neurons along with their connections are randomly removed from the neural network. Ignoring, or "dropping out," individual neurons may help avoid over-adaptation and over-fitting. Each iteration results in the development of a new sub-network with different neurons than the previous iteration. A collection of sub-networks emerges from such a method, which has a better chance of capturing random phenomena in data than a single robust network. When using this method, the parameter, which determines the probability of selecting some neurons, must be a set of sub-networks that have a greater chance of catching random phenomena in the data than a single robust network.

Kernel regularize: - Regularizes allow to apply penalties on layer parameters or layer activity during optimization. These penalties are summed into the loss function that the network optimizes. In our models, we have used kernel regularize l2. L2 regularization reduce overfitting by penalizing the model's weights so that the weights take only small values.

Besides in this study we tested different dropout rate but our model was effective in 0.39 dropout rate and reduced overfitting curve in L2 kernel regularize.

4.2.4 Data cleaning

After the data is collected the next step is specifying the software tool and pre-processing becomes the primary concern of this research. As we have discussed in the previous section the data is collected from different sources and we use different data cleaning mechanisms like stop word removal, punctuation and numbers removal, titles and abbreviations expanding and stemming into the word stem also the parts of the data cleaning. The data cleaning step's main aim was to prepare the data for training and testing.

4.3 Evaluation

4.3.1 Performance Metrics

We choose performance metrics and confusion metrics to measure the model performance using standard classification metrics such as accuracy, precision, recall, and f1-score (Maslej et al. 2020). For multiple classification problems, such metrics are simple to obtain and can be computed as follows:

Accuracy: Is calculated as the sum of correct classifications divided by the total number of classifications. The mathematical expression is given below.

$$\text{Accuracy} = \frac{\text{TP + TN}}{\text{TP + FP + FN + FP}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

Equation 4-3 Equation of Accuracy

Precision: Is a measure of the true positive among all positives. The mathematical expression is given below

$$\text{Precision} = \frac{TP}{TP+FP} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4)$$

Equation 4-4 Equation of Precision

Recall: commonly called sensitivity, corresponds to the true positive rate of the considered class. The mathematical expression is given below

$$Recall = \frac{TP}{TP + FN} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (5)$$

Equation 4-5 Equation of Recall

F1-score: Is the weighted average the precision and recall. The mathematical expression is given below.

$$F1 - Score = \frac{2 * (Precision * Recall)}{Precision + Recall} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

Equation 4-6 Equation of F1-Score

Macro Average: Is just the average of the precision, recall and F1 score of the model on different classes in order to get macro precision, macro recall and macro F1 score respectively. Mathematical expression is given below.

$$\text{Macro average Precision} = \frac{\text{P1} + \text{P2} + \ldots + \text{PN}}{\text{N}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{10})$$

Equation 4-7 Equation of Macro average Precision

$$\text{Macro average Recall} = \frac{\text{R1} + \text{R2} + \ldots + \text{RN}}{\boldsymbol{N}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{11})$$

Equation 4-8 Equation of Macro average Recall

$$\text{Macro average F1score} = \frac{\text{F1Score1} + \text{F1Score2} + \ldots + \text{F1ScoreN}}{\text{N}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{12})$$

Equation 4-9 Equation of Macro average F1Score

Weighted Average: Precision, recall and F1score is multiplied by total number of sample of each class (Y) then divided by total number of samples to get weighted precision, weighted recall and weighted F1score respectively. Mathematical expression is given below.

$$\text{Weighted average Precision} = \frac{\text{P1} * \text{Y1} + \text{P2} * \text{Y2} + \ldots + \text{P(N} *)\text{YN}}{\text{Y1} + \text{Y2} + \ldots + \text{YN}} \ldots\ldots\ldots\ldots\ldots\ldots (13)$$

Equation 4-10 Equation of Weighted average precision

$$\text{Weighted average Recall} = \frac{\text{R1} \ * \text{Y1} \ + \text{R2} \ * \text{Y2} + \ldots + \text{R(N} *\text{)YN}}{\text{Y1} \ + \text{Y2} + \ldots + \text{YN}} \ldots \ldots \ldots \ldots \ldots (14)$$

Equation 4-11 Equation of Weighted average Recall

$$\text{Weighted average F1score} = \frac{\text{F1score1} \ * \text{Y1} \ + \text{F1score2} \ * \text{Y2} + \ldots + \text{F1scoreN} \ * \text{YN}}{\text{Y1} \ + \text{Y2} + \ldots + \text{YN}} \ldots \ldots (15)$$

Equation 4-12 Equation of Weighted average F1Score

where:

True Positive (TP): Is an outcome where the model correctly predicts the positive class

True Negative (TN): Is an outcome where the model correctly predicts the negative class

False Positive (FP) Is an outcome where the model incorrectly predicts the positive class

False Negative (FN): Is an outcome where the model incorrectly predicts the negative class.

The confusion matrix also the other evaluation matrix of this study. Such metrics could be calculated for each class to be used in the multi-label classification based on actual labels with a predicated label in the dataset sentence. The experiment was done on three deep learning algorithms in three different experiments within specified constant and variable hyperparameters. The comparison result of the experiment from those approaches was analyzed both in-text interpretation and graphical analysis. The cost of training is calculated using Mean Square Error (MSE). We also use the confusion matrix to calculate the model's classification results. The experiment was done within a given dataset that is split ration from the total dataset within a ratio of 0.8 for training 0.1 for testing 0.1 for validation purposes.

Table 4:3 Hyperparameters Effect During Three Experiments

Experiments	Constant Hyperparameters	Variable Hyperparameters	LSTM Acc.	CNN Acc.	Bi-LSTM Acc.
Experiment 1	Vocab size = 100,000 Padding=40 Embedding layer = 40 Split ratio = 0.1	Random state=0 Optimizer = SGD Activation = tanh Batch size=32 Epochs =10 Dropout=0.2 Number of neurons =100	80%	92%	91%
Experiment 2		Random state=0 Optimizer = NAdam Activation = relu Batch size=16 Epochs =10 Dropout=0.2 Number of neurons =60	72%	84%	90%
Experiment 3		Random state=42 Optimizer = Adam Activation = gelu Regularize =l2(0.001) Batch size=8 Epochs =20 Dropout=0.39 Number of neurons =40	87%	93%	96%

4.3.2 Experimental Result of CNN Model

The automatic word sense disambiguation model was done by the convolutional neural network by applying the above hyperparameters results analyzed as follows. The model performance was measured based on the evaluation metrics and confusion matrix. From the three experiments, the one which outperforms training accuracy and minimum loss with average evaluation matrix presented. The following curve(right) shows that training accuracy and validation accuracy and curve(left) located the loss graph of training and validation. The training early stopped at epoch 10 as shown in the figure.

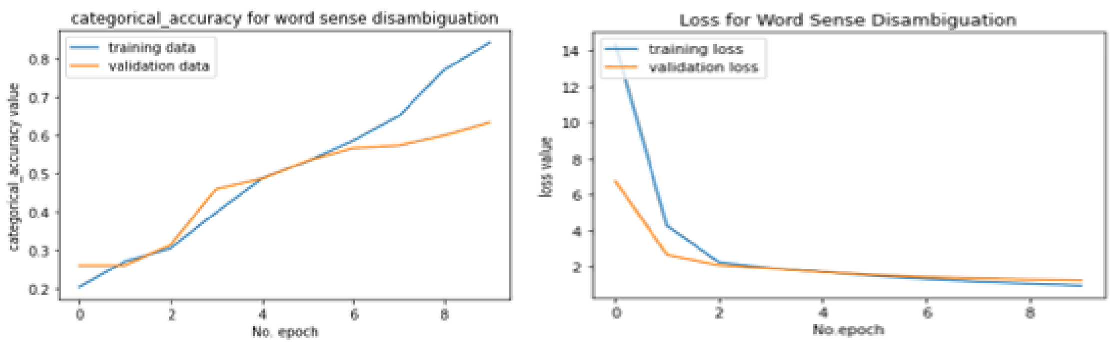


Figure 4:1 Training Accuracy curve(right) with Loss of CNN curve(left) Model

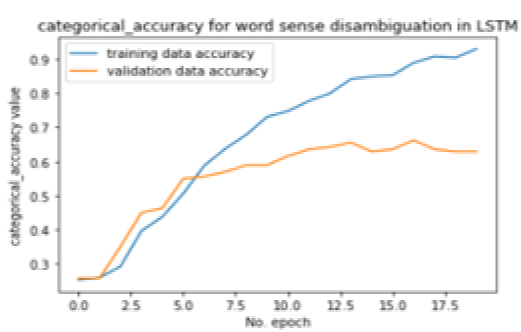
As shown from the above figure the model has no overfitting problem since we used model regularization and dropout rate balancing during the third experiment. This experiment has outperformed result from the remaining two which is attached at appendix I. The overall performance of the CNN model on Amharic word sense disambiguation problem classification by average evaluation metrics during three experiments have represented a table. The results represented in table 4:4 below show average evaluation metrics of CNN model performance in three different experiments.

Table 4:4 Average Evaluation Matrix of CNN model

Classes	*Precision*	*Recall*	*F1-score*	*Support*
0	0.90	0.90	0.90	10
1	0.96	0.92	0.93	9
2	0.92	0.78	0.84	9
3	0.96	0.85	0.90	9
4	0.59	0.59	0.59	9
5	1.00	1.00	1.00	9
6	0.90	1.00	0.95	8
7	0.96	1.00	0.98	9
8	0.82	0.95	0.87	19
Accuracy			0.89	91
Macro avg	0.89	0.89	0.89	91
Weighted avg	0.88	0.89	0.89	91

4.3.3 Experimental Result of LSTM Model

The second experiment done by LSTM model on three times within specified hyperparameter setup. The experimental result which outperforms on Amharic word sense disambiguation model analyzed as follows. The remaining two experiments attached on appendix I. The following graph shows the training and validation accuracy curve(left) with loss curve(right) of the word sense disambiguation problem during the experiment.



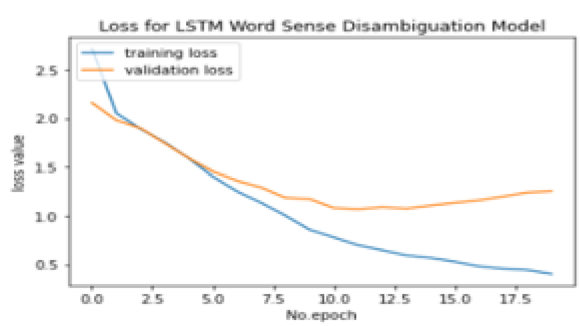


Figure 4:2 Training Accuracy of LSTM curve(left) with Loss of LSTM curve(right)

Rather than the previous experiment by LSTM model in this experiment, there is a change in LSTM model performance as represented in the above figure.

Our LSTM model is evaluated based on performance metrics and a confusion matrix like that of other models. The table shows the average performance metrics of LSTM model on three different experiments.

Table 4:5 Overall average performance of LSTM model by Evaluation matrix

Classes	*Precision*	*Recall*	*F1-score*	*Support*
0	0.97	1.00	0.98	10
1	0.80	0.78	0.79	9
2	0.78	0.89	0.84	9
3	0.78	0.78	0.78	9
4	0.87	0.74	0.80	9
5	1.00	1.00	1.00	8
6	0.86	0.96	0.91	9
7	0.81	0.89	0.84	9
8	1.00	0.91	0.95	19
Accuracy			0.88	91
Macro avg	0.88	0.88	0.87	91
Weighted avg	0.89	0.88	0.88	91

Relatively LSTM model have a problem of overfitting from the two models which means CNN and Bi-LSTM as shown in figure 4:2 the validation loss higher than training loss. It causes the model is not generalized to be able to work on real life dataset.

4.3.4 Experimental Result of Bi-LSTM model

During the experiment the training loss and validation loss of Bi-LSTM model represented by the following graph. During the experiment, the training data and validation data distribution in solving Amharic word sense disambiguation problem one outperforms the three experiments presented by the following graph.

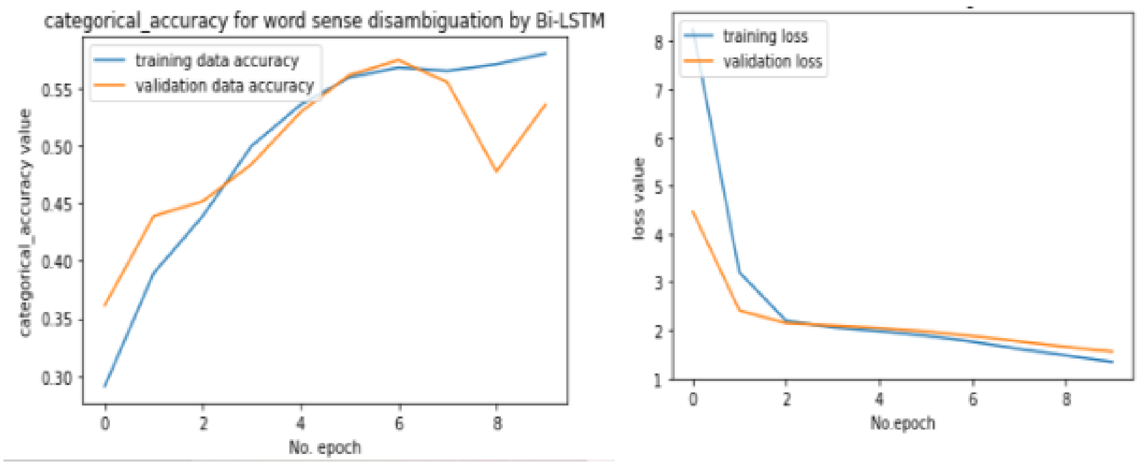


Figure 4:3 Training Accuracy (left) with Loss of Bi-LSTM (right) Model

The experimental result of Bi-LSTM model on this experiment was very effective. As shown in the above figure the model has no overfitting problem. Since we used a model regularize and smaller batch size comfortable for our small dataset.

Lastly, the following table shows the overall average performance of Bi-LSTM model on Amharic word sense disambiguation problem.

Table 4:6 Average Evaluation matrix of Bi-LSTM Model on three Different Experiment

Classes	*Precision*	*Recall*	*F1-score*	*Support*
0	1.00	0.90	0.94	10
1	1.00	0.66	0.73	9
2	0.76	0.81	0.85	9
3	0.84	0.81	0.89	9
4	0.91	0.96	0.93	9
5	0.93	1.00	0.96	9
6	0.93	1.00	0.90	8
7	0.87	0.96	0.96	9
8	0.93	0.96	0.97	19
Accuracy			0.93	91
Macro avg	0.93	0.93	0.93	91
Weighted avg	0.93	0.93	0.93	91

4.3.4 Mean Square Error

Kim, Kim, and others (2017)MSE is a popular method for calculating how far an estimator deviates from the target value. The average of all squared errors of each pattern is the MSE of a dataset.

Let **p** be the predicted value and **a** be the actual value for the ith example, given **m** patterns in the dataset. MSE is calculated using the following equation.

$$MSE = \sum_{i=0}^{n}(pi - ai)2 \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (10)$$

Equation 4-13 Equation of Mean Square Error

The following confusion matrix result gotten from Bi-LSTM experiment. Based on the result the model missed some values when it predicates related to the actual value.

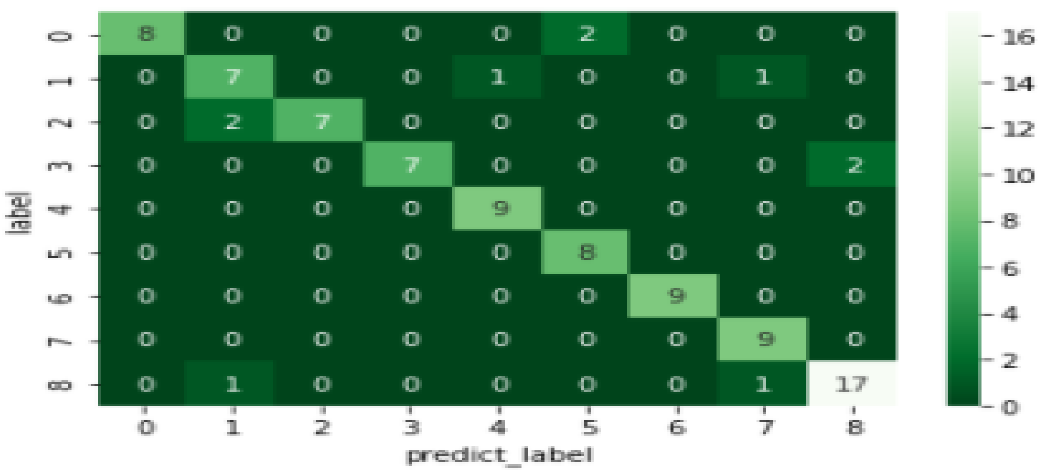


Figure 4:4 confusion matrix of Bi-LSTM Model

Based on the figure above mean square error of Bi-LSTM model calculated as follows:

If **pi** represents the predicate value and **ai** represents each actual value at class I from total test dataset for each class is 91 in this confusion matrix.

Therefore,

$MSE = \sum_{i=0}^{n}(pi - ai)^2$

$= (8-8)^2+(7-10)^2+(7-7)^2+(7-7)^2+(10-9)^2+(8-10)^2+(9-9)^2+(9-11)^2+(17-19)^2$

$= (0+9+0+0+1+4+0+4+4)$

$= 22/91$

$= 0.24$

Based on the mean square error from the actual value mostly missed by the model was antonymy relations. Because of some ambiguous words haven't antonymy. Next to this, hypernymy and holonomy relation were ambiguous for the model. For example, ከስለት ነገሮች አንዱ የሆነውን የቢላ ጫፍ በሞረድ ሳለ።/kesileti negerochi ānidu yehonewini yebīla ch'afi bemoredi sale/ the one ambiguous for the model and it classified in one experiment at hypernymy and in some experiment at class of meronymy. To correct this sentence, we rewrite as ከስለት ነገሮች አንዱ የሆነውን ቢላ በሞረድ ሳለ/kesileti negerochi ānidu yehonewini bīla bemoredi sale/ for hypernymy ከቢላዋ አካል ከፍሎች አንዱ የሆነውን የቢላዉን ጫፍ በሞረድ ሳለ/kebīlawa ākali kefilochi ānidu yehonewini yebīlawuni ch'afi bemoredi sale/ for meronomy.

4.4 Discussion

In this section, we measured the overall performance of the model using a testing dataset by using different evaluating techniques. The experiment was done on three deep learning algorithms in three different experiments with constant and variable hyperparameters. The comparison result found from those approaches was analyzed in both word interpretation and graphical analysis. Then, average accuracy was analyzed in a single table by evaluation metrics.

Experiment 1

During experiment 1 all the three deep learning algorithms tested within the specified constant hyperparameter setup using padding, embedding, split ratio, and vocabulary size. Specifically, from the variable hyperparameter optimizer = SGD, activation = tanh, batch size = 32, random state = 0, number of neurons = 100. In this experiment since we were using tanh activation function but this activation function has a problem of vanishing gradient around -1 and 1. The vanishing gradient problem restricts the model to go global minima. SGD has a high variance in model parameters and to get the same convergence as gradient descent needs to slowly reduce the value of learning rate where the activation function problem affects the results of this experiment. Besides in this experiment especially LSTM (80%) and Bi-LSTM (91%) achieved low performance related to CNN (92%).

Experiment 2

During this experiment rather than the constant hyperparameter specified in experiment 1 its variable hyperparameters were random state=0, Optimizer = NAdam, activation = relu, batch size=16, epochs =10, dropout=0.2, number of neurons =60. The activation function in this experiment was relu have a problem of dead state around 0 and 1. Still, the batch size in this experiment used was largely related to the dataset. That is why the model performance has gone to minimum related to other experiments. The dropout rate also factor affects our experiment. In this experiment, since we used 0.2 dropout rate is not comfortable for our problem. The activation function was relu have given 0 for all negative value and died sate problem. Besides this experiment was not achieved a good result in the three language models related to the other experiments.

Experiment 3

During this experiment the constant hyperparameters were specified in both experiments 1 and 2. But the variables were random state=42, optimizer = Adam, activation = gelu, regularize =l2(0.001), batch size=8, epochs =20, dropout=0.39, number of neurons =40. Since Adam optimizers used in this experiment have a very good learning rate rather than the others. The activation function used in this experiment was gelu(gaussian rectified linear unit) has a potential to learn within mini batch data by solving the problem of dead state problem in relu. The random state is used to regularize and minimize overfitting problem of the graph from 0 to increase 42. The batch size specified in this experiment also comfortable for the models because of the dataset was small. Besides in this experiment relatively all algorithms achieved a good result like LSTM (87%), CNN (93%), Bi-LSTM (96%).

The first experiment was done by convolutional neural network and achieved an average accuracy of 89% without model overfitting problem.

The second experiment was done by LSTM model for the Amharic word sense disambiguation dataset within average accuracy of 88%. The researchers observed from this model was its learning accuracy increases in higher epochs because the model contains only single direction neurons(Maslej Krešňáková et al., 2020). The last experiment was done on Bi-LSTM model which has scored a state-of-the-art result, especially in experiment three 96% and an average 93% accuracy scored. From those three deep learning language models, Bi-LSTM has higher accuracy than the others in three different experiments on word sense disambiguation problems that is why this study used the method to solve the problem.

Before a time, the models have an overfitting problem. To solve this problem, we used the model to regularize by the model (regularize l2) and balancing the dropout rate since it determines which neuron is idle and active at a time.

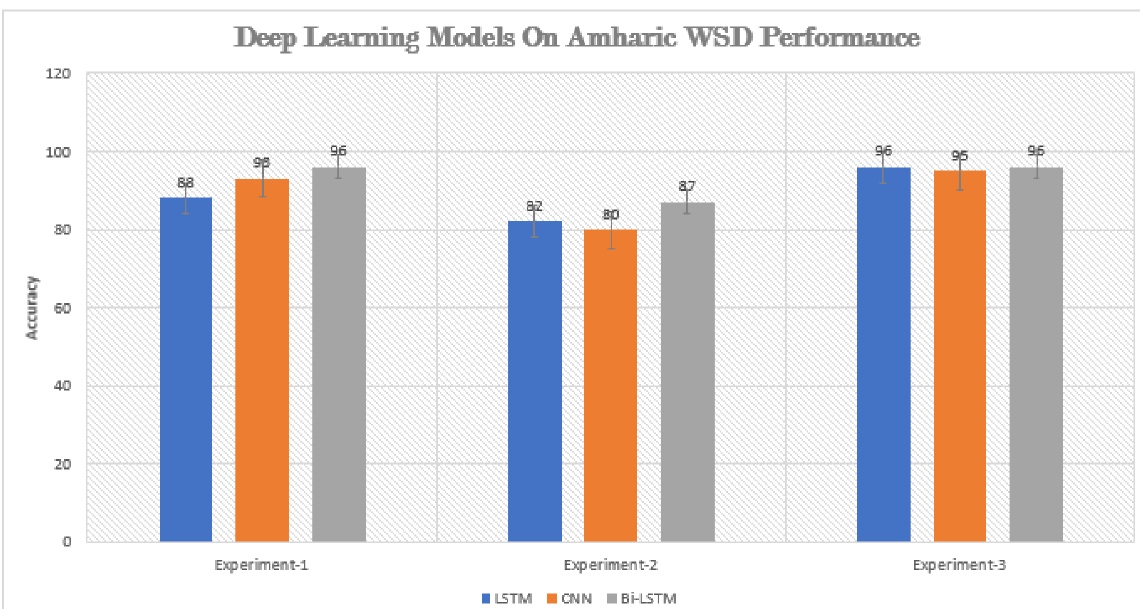


Figure 4:4 Comparison Result of Three Language Models

As shown in figure 4:4 the overall result of the three deep learning language models in three different Experiments. Bi-LSTM was higher performance related to other models for three different experiments. Because of it has double neurons and can learn in both directions backward and forward propagation(Maslej Krešňáková et al., 2020). In addition to, our dataset is low it needs a small batch size to learn every feature effectively. The remaining experimental result which has low performance related to presented above attached in appendix A. Based on our experiment result Bi-LSTM achieved a state-of-the-art result so it's selected as our model to solve the Amharic word sense disambiguation problem less than 16 batch sizes. The performance of the model related to the previous work analyzed as follows.

As tables 4:7 represented below the performance of our Bi-LSTM model and previous works on Amharic WSD our model has a 16% improvement from the previous one and contains four-word relations which are not covered by the previous authors. The research comparison has been done due to problem similarity but the model and dataset strictly differ.

Table 4:7 The Comparison Results in the Previous Amharic WSD Model and our Model.

Authors	Approach	Size of dataset	Accuracy
(Mieraf, 2020)	Knowledge base (context overlap and augmented semantic space)	17 words and 350 sentences for testing only	80% accurate
Our Experiment	Deep learning (Bi-LSTM)	159 words, 460 glosses, 1350 synset, and 2164 sentences for testing and training	96% accurate

Chapter Five

5 Conclusion and Recommendation

5.1 Conclusion

Word sense disambiguation (WSD) means selecting the right sense based on the surrounding context. Word sense disambiguation is done multiple times and effective models build in reach resource languages like English. In such languages word sense disambiguation model builds by contextual embedding without WordNet because of billions of SemCor loaded in the web. Amharic, Arabic, Hebrew, etc. low resource languages have no effective word sense disambiguation models and difficult two implement other high resource language modes because of their morphological complexity and a derivational variant of words in the language structure. Even if some researchers did Amharic word sense disambiguation model. Those weren't cover the holonomy, antonym, homonymy, and troponymy relation of the word in their dataset to disambiguate ambiguous words. And also, the researchers didn't implement deep learning algorithms those handle effective contexts based on the surrounding context rather than shallow machine learning approaches. This research was done by deep learning approach rather than the previous authors within considering the remaining four relations of the word listed above.

The research collected 159 ambiguous words, 460 glosses, and 2164 sentences in different sources. And then we are designing WordNet's based on their relationships within synset and ambiguous words a lexical database. We developed three different deep learning models such as LSTM, CNN, and Bi-LSTM models trained by the dataset sentence within specified Hyperparameter setup in three experiments for each model. The trained model predicts the relation of the word or class and then if it is classified as ambiguous fetches its gloss from the WordNet. The model performance measured in confusion metrics and performance metrics like accuracy, precision, recall, and f1-score. Based on the experimental result LSTM, CNN and Bi-LSTM achieved 88%, 89%, and 93% average accuracy respectively during the three experiments. Based on the experimental result Bi-LSTM achieved the state-of-the-art result rather than the other models. Besides we used a

Bi-LSTM model for solving Amharic ambiguous word problems. Lack of resources and datasets were the main challenges of this study.

5.2 Contribution of the Study

The major contribution of this study includes:

- Previously authors solve the word sense disambiguation problem by using machine learning and knowledge-based approaches. However, this study designs an automatic Amharic word sense disambiguation model by deep learning approach by using WordNet would be the first model.
- We designed three different text classification models such as LSTM, CNN, and Bi-LSTM and analyzed in three different hyperparameter setup. From those Bi-LSTM model achieved state-of-the-art result to solve Amharic word sense disambiguation problem. The model also includes antonym, troponomy, holonomy, and homonym relations or synsets of ambiguous words not covered by the other researchers.
- The dataset prepared for this research 2164 sentences and Amharic WordNet which has 416 gloss in 159 ambiguous words by itself because of containing 8-word relations that were useful for disambiguating ambiguous words for different NLP applications.

5.3 Future Work

The model designed in this study is Automatic Amharic word sense disambiguation by using a deep learning approach with WordNet. A word has multiple meanings, then detecting the meaning based on the context is known as WSD. The study includes 8-word relation of a single word to disambiguate an ambiguous word. However, the word meronomy relations expands furtherly it includes only one meronomy relation. The substance of and entailment members of meronomy were not handled in this study. So, we recommended the following points in the study:

- Designing automatic Amharic word sense disambiguation model which includes the two meronomy relations the substance of and entailment.
- Designing Automatic word sense disambiguation model by using contextual embedding by using BERT without WordNet by applying transfer learning.

References

Abate, S. T., & Menzel, W. (2007). Syllable-based speech recognition for Amharic. *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, 33–40.

Abid, M., Habib, A., Ashraf, J., & Shahid, A. (2018). Urdu word sense disambiguation using machine learning approach. *Cluster Computing*, *21*. https://doi.org/10.1007/s10586-017-0918-0

Alkhatlan, A., Kalita, J., & Alhaddad, A. (2018). Word Sense Disambiguation for Arabic Exploiting Arabic WordNet and Word Embedding. *Procedia Computer Science, 142*, 50–60. https://doi.org/10.1016/j.procs.2018.10.460

Amare, G. (2001). Towards the analysis of ambiguity in Amharic. *Journal of Ethiopian Studies*, *34*(2), 35–56.

Assabie, M. A. and Y. (2014). Development of Amharic Morphological Analyzer Using Memory-Based Learning. *9th International Conference on NLP, PolTAL 2014*, *32*, 1–13. https://doi.org/10.1007/978-3-319-10888-9

Bala, P. (2013). Knowledge Based Approach for Word Sense Disambiguation using Hindi Wordnet. *International Journal of Engineering and Science*, *2*(4), 36–41.

Banerjee, S., & Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *2276*, 136–145. https://doi.org/10.1007/3-540-45715-1_11

Basile, P., De Gemmis, M., Lops, P., & Semeraro, G. (2008). Combining knowledge-based methods and supervised learning for effective Italian word sense disambiguation. *Semantics in Text Processing. STEP 2008 Conference Proceedings*, 5–16.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166.

Bouhriz, N., Benabbou, F., & Lahmar, E. H. Ben. (2016). Word sense disambiguation approach for Arabic text. *International Journal of Advanced Computer Science and Applications*, *7*(4), 381–385.

Braz, F. A., Silva, N. C., de Campos, T. E., Chaves, F. B. S., Ferreira, M. H. P., Inazawa, P. H. G., Coelho, V. H. D., Sukiennik, B. P., Almeida, A. P. G. S., Vidal, F. B., Alves Bezerra, D., Gusmão, D. B., Ziegler, G. G., Fernandes, R. V. C., Zumblick, R., & Hartmann Peixoto, F. (2018). Document classification using a Bi-LSTM to unclog Brazil's supreme court. *ArXiv*.

Braz, F., Silva, N., de Campos, T., Chaves, F., Ferreira, M., Inazawa, P., Coelho, V., Sukiennik, B., Almeida, A., Vidal, F., Bezerra, D., Gusmao, D., Ziegler, G., Fernandes, R., Zumblick, R., & Peixoto, F. (2018). *Document classification using a Bi-LSTM to unclog Brazil's supreme court*.

Chaplot, D. S., & Bhattacharyya, D. P. (2014). Literature survey on unsupervised word sense disambiguation. *IIT Bombay, May*, *7*, 39.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *ArXiv Preprint ArXiv:1406.1078*.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *ArXiv Preprint ArXiv:1412.3555*.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(ARTICLE), 2493–2537.

Daelemans, W., den Bosch, A., & others. (2005). *Memory-based language processing*.

Cambridge University Press.

Devendra, S. (2014). Literature Survey on UnsupervisedWord Sense Disambiguation. . *Indian Institute of Technology, Computer Science and Engineering. Bombay: Indian Institute of Technology.*, *8*, 5–16.

Du, J., Qi, F., & Sun, M. (2019). *Using BERT for Word Sense Disambiguation*. *1*.

Dureti, S. B. (2017). *A Generic Approach towards all Words Amharic Word Sense Disambiguation*. Addis Ababa University.

Fellbaum, C. (2010). WordNet. In *Theory and applications of ontology: computer applications* (pp. 231–243). Springer.

Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*, *3*(Aug), 115–143.

Giyanani, R. (2013). A survey on word sense disambiguation. *IOSR Journal of Computer Engineering (IOSR-JCE)*, *14*, 30–33.

Graves, A. (2012). Supervised sequence labelling. In *Supervised sequence labelling with recurrent neural networks* (pp. 5–13). Springer.

Hadiwinoto, C., Ng, H., & Gan, W. (2019). *Improved Word Sense Disambiguation Using Pre-Trained Contextualized Word Representations*.

Heo, Y., Kang, S., & Seo, J. (2020). Hybrid Sense Classification Method for Large-Scale Word Sense Disambiguation. *IEEE Access*, *PP*, 1. https://doi.org/10.1109/ACCESS.2020.2970436

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Huang, L., Chi, S., Qiu, X., & Huang, X. (2019). *GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge*.

Hung, J. C., Wang, C.-S., Yang, C.-Y., Chiu, M.-S., & Yee, G. (2005). Applying word sense disambiguation to question answering system for e-learning. *19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA Papers)*, *1*, 157–162.

Jang, B., Kim, M., Harerimana, G., Kang, S., & Kim, J. W. (2020). Bi-LSTM Model to Increase Accuracy in Text Classification : Combining Word2vec CNN and Attention Mechanism. *Applied Sciences*, *10*, 1–14.

Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015). An Empirical Exploration of Recurrent Network Architectures. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd International Conference on Machine Learning* (Vol. 37, pp. 2342–2350). PMLR. http://proceedings.mlr.press/v37/jozefowicz15.html

Kågebäck, M., & Salomonsson, H. (2016). *Word Sense Disambiguation using a Bidirectional LSTM*.

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. *ArXiv Preprint ArXiv:1404.2188*.

Kim, J., Kim, H., & others. (2017). An effective intrusion detection classifier using long short-term memory with gradient descent optimization. *2017 International Conference on Platform Technology and Service (PlatCon)*, 1–6.

Kleenankandy, J., & KA, A. N. (2020). An enhanced Tree-LSTM architecture for sentence semantic modeling using typed dependencies. *Information Processing & Management*, *57*(6), 102362.

Kumar, P. (2020). Word sense disambiguation for Punjabi language using deep learning techniques. *Neural Computing and Applications*, *32*. https://doi.org/10.1007/s00521-019-04581-3

Kumar, R., Khanna, R., & Goyal, V. (2012). A review of literature on word sense disambiguation. *International Journal of Engineering Sciences*, *6*, 224–230.

Kumar, S., Jat, S., Saxena, K., & Talukdar, P. (2019). *Zero-shot Word Sense Disambiguation using Sense Definition Embeddings*. https://doi.org/10.18653/v1/P19-1568

Li, B., Thomas, G., Xu, Y., & Rudzicz, F. (2020). Word class flexibility: A deep contextualized approach. *ArXiv Preprint ArXiv:2009.09241*.

Li, C. W. (2017). Automatic Synonym Extraction Using Word2Vec and Spectral Clustering. *Proceedings of the 36th Chinese Control Conference July 26-28, 2017, Dalian, China.*, 1.

Lo, Ê., Chen, É., Lin, Q., & Hsieh, S.-K. (2008). Automatic labeling of troponymy for Chinese verbs. *Proceedings of the 20th Conference on Computational Linguistics and Speech Processing, ROCLING 2008*.

Loureiro, D., Pilehvar, M. T., Rezaee, K., & Camacho-Collados, J. (2020). Language models and Word Sense Disambiguation: An overview and analysis. *ArXiv*.

Lu, Y., & Salem, F. M. (2017). Simplified gating in long short-term memory (lstm) recurrent neural networks. *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 1601–1604.

Ma M., & M. P. (2003). Temporal relations in visual semantics of verbs. *In Proc. of The Fourteenth Irish Conference on Artificial Intelligence and Cognitive Science (AICS-03)*, 128–133.

Maslej Krešňáková, V., Sarnovsky, M., Butka, P., & Machova, K. (2020). Comparison of Deep Learning Models and Various Text Pre-Processing Techniques for the Toxic Comments Classification. *Applied Sciences*, *10*, 8631. https://doi.org/10.3390/app10238631

Meng, F. (2020). Graph and Word Similarity for Word Sense Disambiguation. *13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI) Graph*, 1114–1118.

Miangah, T. M., & Khalafi, A. D. (2005). Word sense disambiguation using target language corpus in a machine translation system. *Literary and Linguistic Computing*, *20*(2), 237–249.

Mieraf, M. (2020). *"Word Sense Disambiguation for Amharic Sentences using WordNet Hierarchy"Bahir Dar, Ethioipia: Unpublished Master thesis, Bahir Dar University*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *ArXiv Preprint ArXiv:1301.3781*.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, *3*(4), 235–244.

Mulat, G. (2020). *"Amharic WordNet construction Using Word Embedding"Addis Ababa, Ethiopia: Unpublished Master thesis, Addis Ababa University*.

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, *41*(2), 1–69.

Ng, H. T., & Zelle, J. (1997). Corpus-based approaches to semantic interpretation in NLP. *AI Magazine*, *18*(4), 45.

Nguyen, A., Pham, K., Ngo, D., Ngo, T., & Pham, L. (2021). *An Analysis of State-of-the-art Activation Functions For Supervised Deep Neural Network*.

Ogunfunmi, T., Ramachandran, R. P., Togneri, R., Zhao, Y., & Xia, X. (2019). A primer on deep learning architectures and applications in speech processing. *Circuits, Systems, and Signal Processing*, *38*(8), 3406–3432.

Park, S., Song, K., Ji, M., Lee, W., & Moon, I.-C. (2019). Adversarial dropout for recurrent neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 4699–4706.

Pesaranghader, A., Pesaranghader, A., & Sokolova, M. (2018). *One Single Deep*

Bidirectional LSTM Network for Word Sense Disambiguation of Text Data (pp. 96–107). https://doi.org/10.1007/978-3-319-89656-4_8

Pierpaolo, B., Marco, d. G., Pasquale, L., & Giovanni, S. (2008). Combining Knowledge-based Methods and Supervised Learning for Effective Italian Word Sense Disambiguation. Semantics in Text Processing. . *Venice,Italy: Assosiation for Computational Linguistics.*, *8*, 5–16.

Rajani, M., & Ravi, B. S. (2015). Performance Comparison of Word Sense Disambiguation Approaches for Indian Languages. . *International Advance Computing Conference (IACC).IEEE.*, 166–169.

Roberto, N. (2009). Word Sense Disambiguation: . *ACM Computing Surveys*, *41(2)*, 10–69. https://doi.org/doi:10.1145/1459352.1459355

Sahin, G. (2017). EXTRACTION OF HYPONYMY, MERONYMY, AND ANTONYMY RELATION PAIRS: A BRIEF SURVEY. *International Journal on Natural Language Computing (IJNLC)* , *6*, 1–17.

Sak, H., Senior, A. W., & Beaufays, F. (2014). *Long short-term memory recurrent neural network architectures for large scale acoustic modeling*.

Salem, F. M. (2016). Reduced parameterization in gated recurrent neural networks. *MSU Memorandum*, *7*(11).

Sharma, R., & Bhatia, P. G. (2008). *"Word sense disambiguation for Hindi language"Patiala: India Unpublished master's thesis, Thapar University*.

Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014). Learning semantic representations using convolutional neural networks for web search. *Proceedings of the 23rd International Conference on World Wide Web*, 373–374.

Shree, M. R., & Shambhavi, B. R. (2015). Performance comparison of Word sense disambiguation approaches for Indian languages. *2015 IEEE International Advance Computing Conference (IACC)*, 166–169.

Solomon, A. (2011). Unsupervised Machine Learning Approach For Word Sense Disambiguation To Amharic Words, Addis Ababa: Ethiopia, Unpublished Master's Thesis. *Adiss Ababa University*.

Solomon, M. (2010). Word Sense Disambiguation For Amharic Text: A Machine Learning Approach, Addis Ababa: Ethiopia, Unpublished Master's Thesis,. *Adiss Ababa University*.

Tesema, Y. B. (2016). Word Sense Disambiguation for Amharic Sentences using WordNet Hierarchy. *Addis Ababa University*.

Toruno\uglu-Selamet, D., \.Inceo\uglu, A., & Eryi\ugit, G. (2020). Preliminary Investigation on Using Semi-Supervised Contextual Word Sense Disambiguation for Data Augmentation. *2020 5th International Conference on Computer Science and Engineering (UBMK)*, 337–342.

Vickrey, D., Biewald, L., Teyssier, M., & Koller, D. (2005). Word-sense disambiguation for machine translation. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 771–778.

Viera Maslej-Kreš ˇnáková Peter Butka and Kristína Machová, M. S. *. (2020). Comparison of Deep Learning Models and Various Text Pre-Processing Techniques for the Toxic Comments Classification. *Applied Sciences*, 1–26.

Wassie, G., Ramesh, B. P., Abate, S., & Meshesha, M. (2014). A Word Sense Disambiguation Model for Amharic Words using Semi-Supervised Learning Paradigm. *Science, Technology and Arts Research Journal*, *3*, 147. https://doi.org/10.4314/star.v3i3.25

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

Yarowsky, D. (1992). Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. *COLING 1992 Volume 2: The 15th International*

Conference on Computational Linguistics.

Yesuf, S. H., & Assabie, Y. (2017). Amharic Word Sense Disambiguation Using Wordnet. *The 5 Th International Conference on the Advancement of Science and Technology*, 1.

Ymam, B. (2000). የአማርኛ ሰዋስው የተሻሻለው ሁለተኛው እትም. In *የኢትዮጵያ ቋንቋዎችና ስነ ጽሁፍ.*

Yuan, D., Richardson, J., Doherty, R., Evans, C., & Altendorf, E. (2016). Semi-supervised word sense disambiguation with neural models. *ArXiv Preprint ArXiv:1603.07012*.

Zhou, G.-B., Wu, J., Zhang, C.-L., & Zhou, Z.-H. (2016). Minimal gated unit for recurrent neural networks. *International Journal of Automation and Computing*, *13*(3), 226–234.

Zou, F., Shen, L., Jie, Z., Zhang, W., & Liu, W. (2019). A sufficient condition for convergences of adam and rmsprop. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11127–11135.

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

Appendix I

ባሕር ዳር ዩኒቨርሲቲ

ባሕር ዳር ቴክኖሎጂ ኢንስቲትዩት

ኢንፎርሜሽን ቴክኖሎጂ ትምህርት ክፍል

በባህር ዳር ቴክኖሎጅ ኢንስቲቱት በኢንፎርሜሽን ትምህርት ክፍል ማስተርስ ዲግሪ ማሟያ ለቋንቋ ባለሙያዎች የቀረበ የመረጃ ክለሳና ምደባ

- የአማርኛ የሚያምታቱ ቃላት አገባባዊ ፍች ላይ ያላቸውን ይሁንታ በኢትዮጵያ ቋንቋዎችና ስነ ጽሁፍ ባለሙያዎች የቀረበ ግምገማ

ስለ ትብብርዎ ከልብ እናመሰግናለን!!!

እባክዎ የሚከተለውን ህጎ ደንብ በደንብ ካነበቡ በሁላ ከሰንጠረዡ ውስጥ ያሉትን አረፍተ ነገሮች የያዘዉን ቃል ጽንሰ ሀሳብ በመረዳት ከሚመለከተው ክፍል ይመድቡ።

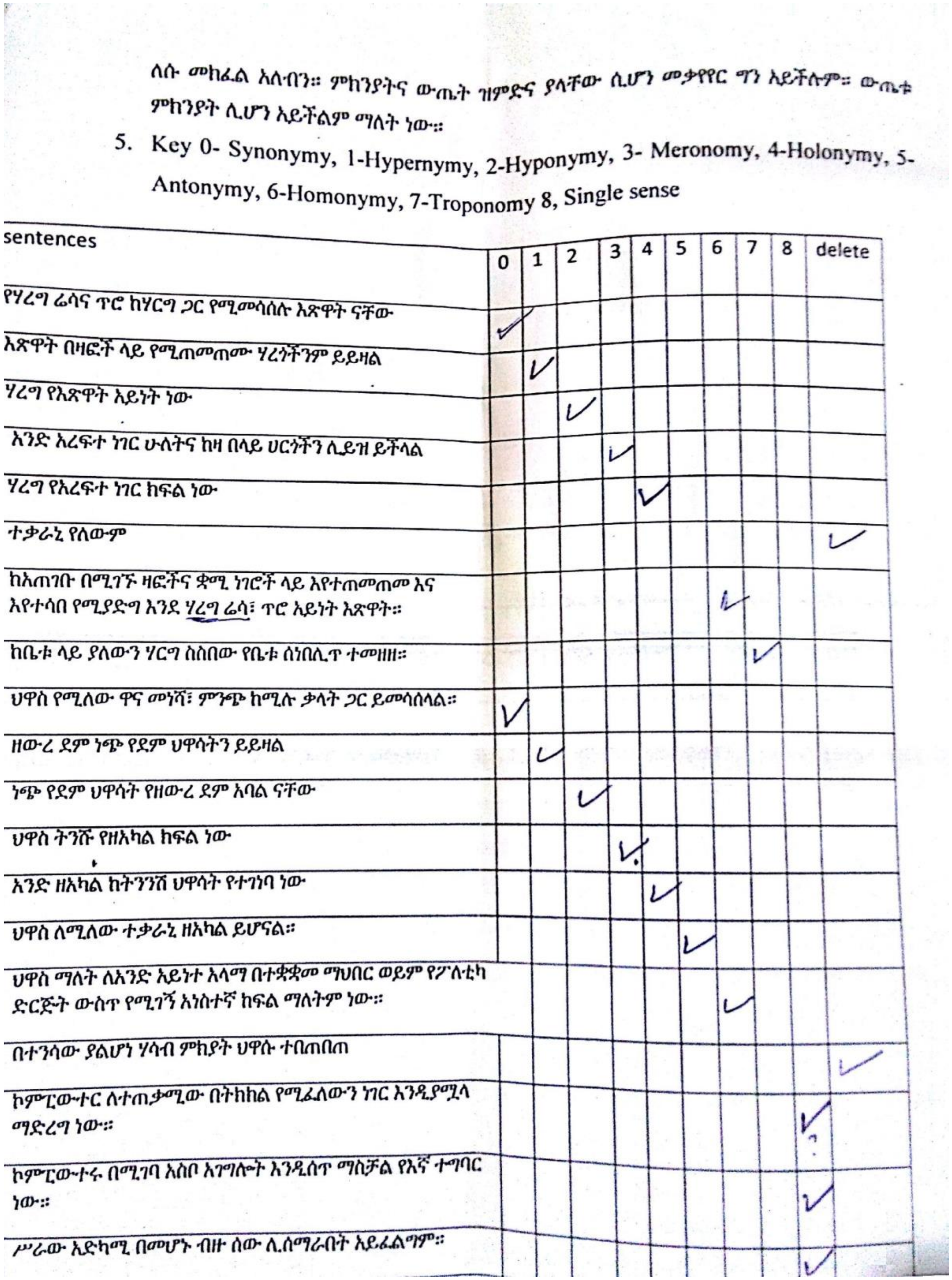
1. በመጀመሪያ አረፍተ ነገሩ የተሰራበትን ዝምድና ምንነት ከ1-4 ከተቀመጠው ማብራሪያ በደንብ ይመልከቱ
2. አረፍተ ነገሮችን በትክክል አንብበው ይረዱት
3. የአረፍተ ነገሩ አወቃቀረ የቋንቋውን ስርአት ካልጠበቀ ሰርዝ/Delete/ የሚለውን ይምረጡ
4. የተቀመጠው አረፍተ ነገር ከተመደበበት ዝምድና ውጭ ነው ብለው ካሰቡ ብቻ ከሚመለከተው ዝምድና ላይ የ ‘X’ ምልክት ያድርጉ

   ልብ ያድርጉ እያንዳንዱ የእርስዎ መልስ በጥናታችን ላይ ከፍተኛ ተጽዕኖ አለው

   3. Hypernymy and Hyponymy relationship- family – member relationship. ISA is-a – member of. ምሳሌ እንስሳ - ውሻ እንስሳ የውሻ ቤተሰብ ሲሆን ውሻ ደግሞ የእንስሳ አባል ነው። በዚህ ዝውድና ላይ እንስሳ- Hypernymy ሲሆን ውሻ- Hyponymy ነው
   4. Meronomy/Holonomy በጠቅላላው እና በከፊል መካከል የፍቺ ግንኙነትን ይወክላል። በዚህ ዝምድና ውስጥ የሙሉው ክፍል ስያሜ Meronomy ተብሎ የሚጠራ ሲሆን የተወሰነው ክፍል ደግሞ Holonomy ተብሎ ይጠራል ፡፡ ምሳሌ ሰው እና እጅ - Meronomy - ሰው and Holonomy - እጅ

      ጥፍር የጣት አካል ነው ›እና‹ ጣት የክንድ አካል ነው የሚሉ ተጨማሪ ምሳሌዎችንም መውሰድ ይቻላል

5. Homonymy- ይህ አይነት የቃል ትርጉም የሚገኘው የቃላቱ አጻጻፍ እና ድምጽ አንድ አይነት ሁኖ ነገር ግን የተለየ ትርጉም ካለው ከመጀመሪያው የተለየው ትርጉም homonymy ይባላል። ለምሳሌ ሳለ ለሚለው ቃል በጉንፋን ምክንያት ኡህ አህ አለ የሚለው የመጀመሪያው ከሆነ ግራፍ በወረቀት ሳለ ማለትም ነው የሚለው Homonymy ይባላል
6. Troponymy - ይህ አይነት የቃላት ዝምድና የቃላቱ ግስ አመክንዮአዊ የግንኙነት ያለው ከሆነ ነው። ለምሳሌ አንድ ሰው የሚያንኮራፋ ከሆነ እንቅልፍ ወስዶታል ማለት ነው ወይም ማንኛውም ነገር ከገዛን ለሱ መክፈል አለብን። ምክንያትና ውጤት ዝምድና ያላቸው ሲሆን መቃየየር ግን አይችሉም። ውጤቱ ምክንያት ሊሆን አይችልም ማለት ነው።
7. Key 0- Synonymy, 1-Hypernymy, 2-Hyponymy, 3- Meronomy, 4-Holonymy, 5-Antonymy, 6-Homonymy, 7-Troponomy 8, Single sense



ለሱ መከፈል አለብን። ምክንያትና ውጤት ዝምድና ያላቸው ሲሆን መቃየር ግን አይችሉም። ውጤቱ ምክንያት ሊሆን አይችልም ማለት ነው።

5. Key 0- Synonymy, 1-Hypernymy, 2-Hyponymy, 3- Meronomy, 4-Holonymy, 5-Antonymy, 6-Homonymy, 7-Troponomy 8, Single sense

sentences	0	1	2	3	4	5	6	7	8	delete
የሃረግ ሬሳና ጥሮ ከሃርግ ጋር የሚመሳሰሉ እጽዋት ናቸው	✓									
እጽዋት በዛፎች ላይ የሚጠመጠሙ ሃረጎችንም ይይዛል		✓								
ሃረግ የእጽዋት አይነት ነው			✓							
አንድ አረፍተ ነገር ሁለትና ከዛ በላይ ሀርጎችን ሊይዝ ይችላል				✓						
ሃረግ የአረፍተ ነገር ክፍል ነው					✓					
ተቃራኒ የለውም										✓
ከአጠገቡ በሚገኙ ዛፎችና ቋሚ ነገሮች ላይ እየተጠመጠመ እና እየተሳበ የሚያድግ እንደ ሃረግ ሬሳ፣ ጥሮ አይነት እጽዋት።							✓			
ከቤቱ ላይ ያለውን ሃርግ ስስበው የቤቱ ሰነበሊጥ ተመዘዘ።								✓		
ህዋስ የሚለው ዋና መነሻ፣ ምንጭ ከሚሉ ቃላት ጋር ይመሳሰላል።	✓									
ዘውረ ደም ነጭ የደም ህዋሳትን ይይዛል		✓								
ነጭ የደም ህዋሳት የዘውረ ደም አባል ናቸው			✓							
ህዋስ ትንሹ የዘአካል ክፍል ነው				✓						
አንድ ዘአካል ከትንንሽ ህዋሳት የተገነባ ነው					✓					
ህዋስ ለሚለው ተቃራኒ ዘአካል ይሆናል።						✓				
ህዋስ ማለት ለአንድ አይነት አላማ በተቋቋመ ማህበር ወይም የፖሉቲካ ድርጅት ውስጥ የሚገኝ አነስተኛ ክፍል ማለትም ነው።							✓			
በተንሳው ያልሆነ ሃሳብ ምክያት ህዋሱ ተበጠበጠ										✓
ኮምፒውተር ለተጠቃሚው በትክክል የሚፈለውን ነገር እንዲያሟላ ማድረግ ነው።									✓	
ኮምፒውተሩ በሚገባ አስቦ አገልሎት እንዲሰጥ ማስቻል የእኛ ተግባር ነው።									✓	
ሥራው አድካሚ በመሆኑ ብዙ ሰው ሊሰማራበት አይፈልግም።									✓	

Figure 4:5 sample dataset reviewed by experts response snapshot

Reference for Reviewing our Dataset

Gashawu Arutie Asaye(PhD)

Assistant Professor of Theoretical and descriptive linguistics at Bahir Dar University

Bahir Dar University

Phone: +251-912-660-825

Email: gashaw.arutie@aau.edu.et

Appendix II

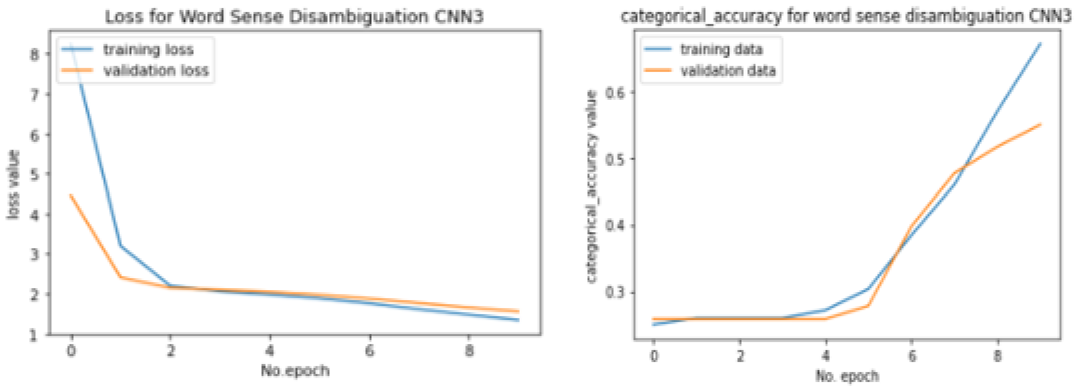


Figure 4:5 Experimental result of loss(left) CNN model Accuracy(right) on experiment 1

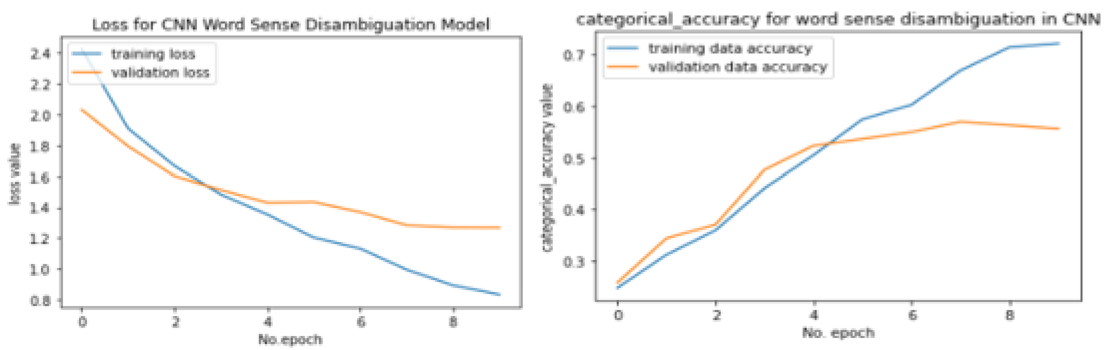


Figure 4:6 Experimental result of loss(left) CNN model accuracy(right) on experiment 2

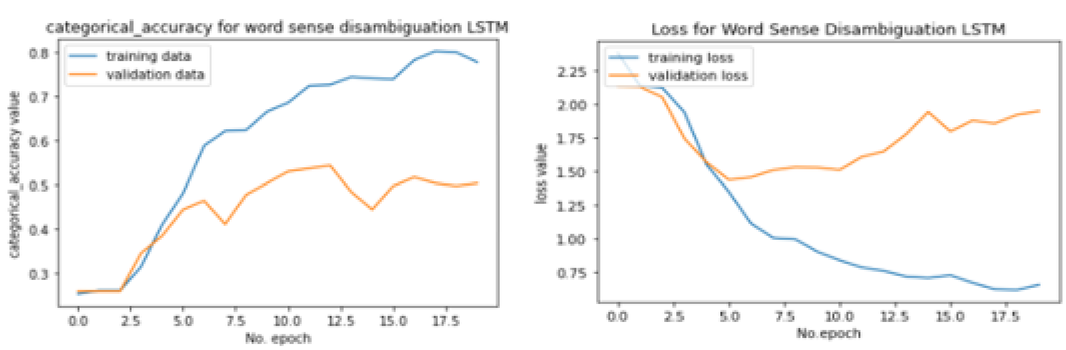


Figure 4:7 Experimental result accuracy(left) of LSTM model loss(right) on experiment 1

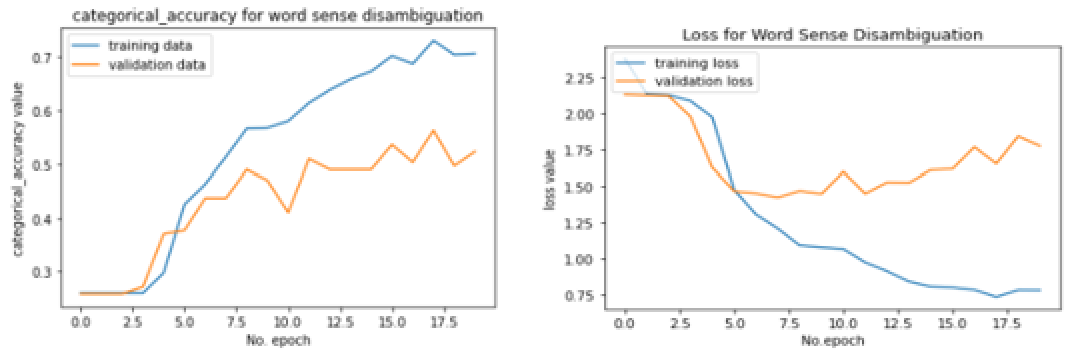


Figure 4:8 Experimental result accuracy(left) model loss(right) on experiment 2

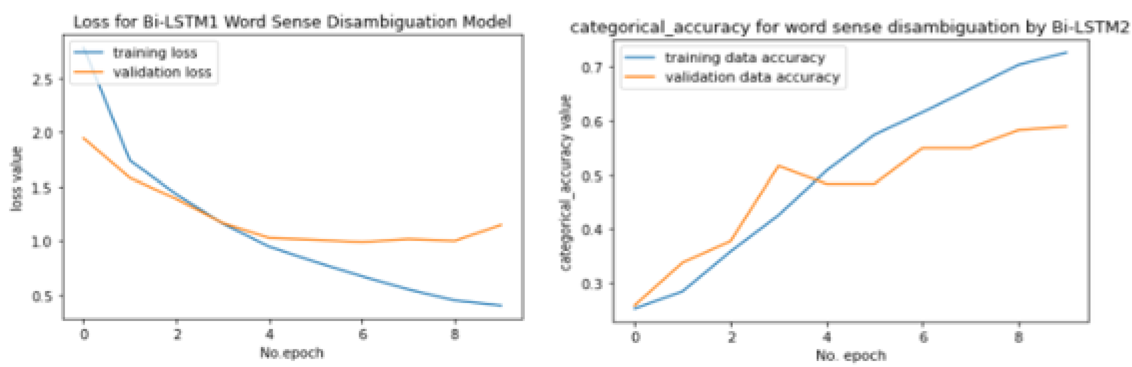


Figure 4:9 Experimental result loss(left) of Bi-LSTM model accuracy(right) on experiment 1

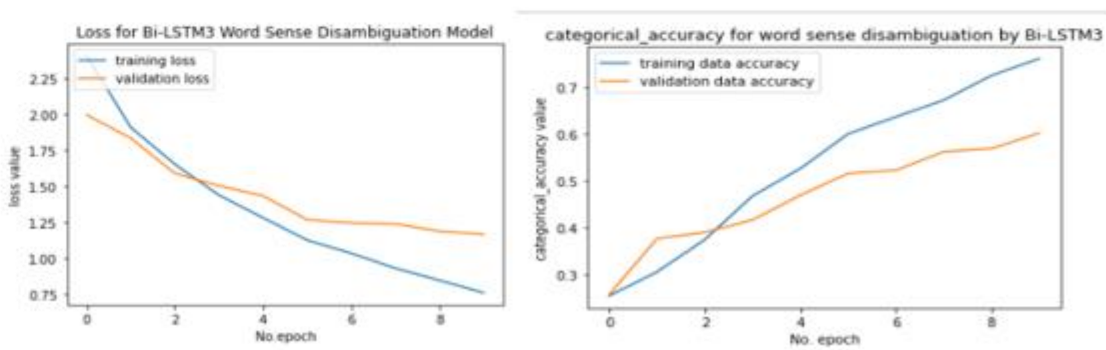


Figure 4:10 Experimental result loss(left) of Bi-LSTM model accuracy(right) on experiment 2

	sentences	label	relation	Num_words_sentences
0	á˜€á˜¨áŒ á˜¬á˜£áŠ" áŒ¥á˜® áŠ¨á˜€á˜-áŒ á˜šá˜ˆá˜£á˜ á˜‰ áŠ¥á…á‹‹á‰µ	0	Synonym	6
1	áŠ¥á…á‹ á‰ á‹áŽ á˜šáŒ á˜ˆáŒ á˜™ á˜€á˜¨áŒŽá‰½áŠ•á˜ á‹-á‹á˜	1	Hypernomy	5
2	á˜€á˜¨áŒ áŠ¥á…á‹‹á‰µ áŠ á‹-áŠá‰µ	2	Hyponomy	3
3	áŠ á˜¨áá‰° á˜á˜ˆá‰µáŠ" áŠ¨á‹ á‰ á˜‹á‹- á˜€á˜-áŒŽá‰½áŠ• á˜Šá‹-á‹ á‰½á˜‹á˜	3	Meronomy	7
4	á˜€á˜¨áŒ áŠ á˜¨áá‰° áŠ-áá˜	4	Holonomy	3
6	áŠ¨áŠ áŒ áŒˆá‰¡ á‰ á˜šáŒˆáŠ™ á‹áŽá‰½áŠ" á‰‹á˜š á‰°áŒ á˜ˆáŒ á˜ˆ á‰°á˜£á‰ á˜šá‹«á‹µáŒ á˜€á˜¨	6	Homonomy	12
7	áŠ¨á‰¤á‰± á‹«á˜ˆá‹áŠ• á˜€á˜-áŒ á˜¥á˜¥á‰ á‹ á‹¨á‰¤á‰± á˜ áŠá‰ á˜ŠáŒ¥ á‰°á˜ˆá‹ˆá‹ˆ	7	Troponomy	7
8	á˜…á‹á˜¥ á˜šá˜ˆá‹ á˜ˆáŠá˜» á˜áŠ•áŒ- áŠ¨á˜šá˜‰ á‰ƒá˜‹á‰µ á˜£á˜ á˜‹á˜	0	Synonym	7
9	á‹ˆá‹á˜¨ á‹°á˜ áŠáŒ- á‹¨á‹°á˜ á˜…á‹á˜£á‰µáŠ• á‹-á‹á˜	1	Hypernomy	6
10	áŠáŒ- á‹¨á‹°á˜ á˜…á‹á˜£á‰µ á‹ˆá‹á˜¨ á‹°á˜ áŠ á‰£á˜	2	Hyponomy	6
11	á˜…á‹á˜¥ á‰µáŠ•á˜¹ á‹ˆáŠ áŠ«á˜ áŠ-áá˜	3	Meronomy	4
12	á‹ˆáŠ áŠ«á˜ áŠ¨á‰µáŠ•áŠ•á˜½ á˜…á‹á˜£á‰µ á‰°áŒˆáŠá‰£	4	Holonomy	4
13	á˜…á‹á˜¥ á˜šá˜ˆá‹ á‰°á‰ƒá˜«áŠ' á‹ˆáŠ áŠ«á˜ á˜†áŠ"á˜	5	Antonomy	5
14	á˜…á‹á˜¥ áŠ á‹-áŠá‰° áŠ á˜‹á˜› á‰ á‰°á‰‹á‰‹á˜ˆ á˜›á˜…á‰ á˜- á–á˜ˆá‰²áŠ« á‹µá˜-áŒ…á‰µ á‹á˜¥áŒ¥	6	Homonomy	11
15	á‰ á‰°áŠ•á˜£ á‹«á˜á˜†áŠ á˜€á˜£á‰¥ á˜áŠ-á‹«á‰µ á˜…á‹á˜¡ á‰°á‰ áŒ á‰ áŒ	7	Troponomy	6
16	áŠ¥á˜¥áŠªá‹«á˜- á‹µá˜¨á˜¥ á‰°á‰†á˜á‰¶ á‰°á‰€á‰€á˜ˆ áŒˆá‰¥á˜¥ á˜¥áŠ•á‹´ á‹‰á˜€ á˜€á˜©á˜- á‰£	0	Synonym	9
17	á˜€á˜©á˜- áŠ áŠ•á‹± áŠ¨áŠ á˜ˆá˜›á‰½áŠ• áŠ á‹¨á˜- á‰¥á˜¨á‰µ á‹á˜¥áŒ¥ á˜šáŠ«á‰°á‰µ	1	Hypernomy	7
18	áŠ á˜ˆá˜› áŠ á‹¨á˜- á‰¥á˜¨á‰µ á˜€á˜©á˜«á˜›áŠ" á‹°áŒ‹á˜›á‹ˆá‰½áŠ• á‹-á‹á˜	2	Hyponomy	6
19	áŠ á‹á˜-áŠ¥ á˜€á˜©á˜- á˜†áŠ'á‰µáŠ•á˜ á‹«áŠ«á‰µá‰³á˜	3	Meronomy	4
20	á‰°á‰€á‰€á˜ˆ áŒˆá‰¥á˜¥ á‹‰á˜€ áŠ¨áŒˆá‰¥á˜¥ á‰°áŒˆáŠˆ	4	Holonomy	5
21	á‹°áŒ‹ á‹¨ á˜€á˜©á˜- á‰°á‰ƒá˜«áŠ'	5	Antonomy	4
22	áŠ¨áá‰° á…á˜€á‹- á˜™á‰€á‰° á‰ƒáŒ á˜Ž á˜€á˜©á˜- á‰£á˜‹á˜	6	Homonomy	6

Figure 4:11 Preprocessed and label sample dataset ready for training