

የኢፌዲሪ የቴክኒክና ሙያ ስልጠና ኢንስቲትዩት
FDRE TECHNICAL & VOCATIONAL TRAINING INSTITUTE

FDRE TECHNICAL & VOCATIONAL TRAINING INSTITUTE

FACULTY OF ELECTRICAL/ELECTRONICS & ICT

DEPARTMENT OF INFORMATION & COMMUNICATION TECHNOLOGY

M.Sc. THESIS

AMHARIC TEXT CHUNCKER USING MACHINE LEARNING

BY

MESELE GAWO EMO

ADVISOR

GETINET YILMA (Ph.D.)

Addis Ababa Ethiopia

June 2026

FDRE TECHNICAL & VOCATIONAL TRAINING INSTITUTE

FACULTY OF ELECTRICAL/ELECTRONICS & ICT

AMHARIC TEXT CHUNCKER USING MACHINE LEARNING

A Thesis Submitted To the Department of Information and Communication Technology, Faculty of Graduate Studies FTVTI

By

Mesele Gawo Emo

ID: - TTMR/156/16

In Partial Fulfillment of the Requirement for the Degree of Master of Science in Information and Communication Technology

ADVISOR

GETINET YILMA (Ph.D.)

June 2026

Addis Ababa Ethiopia

DECLARATION

I hereby declare that this Thesis is my original work and has been completed in accordance with accepted academic and ethical standards. Throughout the preparation of this study, including data collection, data analysis, model development, and documentation, I have followed the required scholarly and technical principles. All sources, ideas, and scholarly materials used in this Thesis have been properly acknowledged through appropriate citations.

This Thesis is submitted in partial fulfillment of the requirements for the Degree of Master of Information and Communication Technology at the FDRE Technical and Vocational Training Institute. A copy of the Thesis is deposited in the Institute library and may be made available to readers and borrowers in accordance with the Institute's regulations.

I further declare that this Thesis has not been submitted, either in whole or in part, to any other institution for the award of any degree, diploma, or certificate.

Brief quotations from this Thesis may be used without special permission, provided that full and proper acknowledgement of the source is given. Permission for extended quotations or reproduction of this Thesis, in whole or in part, may be granted by the Head of the Faculty or Department when such use is considered beneficial for academic and research purposes. In all other cases, permission must be obtained from the author of the Thesis.

Name: Mesele Gawo Emo.

Date:_____ Signature _____

Department: - ICT Information and Communication Technology.

Date _____ Signature: _____

APROVAL PAGE

I hereby certify that I have read and evaluated this Thesis entitled Amharic Text Chunk Using Machine Learning Approach: Prepared under my guidance by Mesele Gawo Emo, ID No TTMR/156/16 I recommend that it be submitted partial fulfilling the thesis Research requirement.

GETINET YILMA (Ph.D.)

Major Advisor  June, 2026

Signature **Date**

As a member of the Board of Examiners of the MSc. Thesis Open Defense Examination, I certify that I have read and evaluated the thesis prepared by Mesele Gawo Emo and examined the candidate. I recommend that the Thesis be accepted as partial fulfilling the Thesis requirements for the degree of Master of Science in Information and Communication Technology.

Dr._____ _____ _____

Chairperson **Signature** **Date**

Dr._____(Ph.D.) _____ _____

Internal Examiner **Signature** **Date**

Dr. _____(Ph.D.) _____ _____

External Examiner **Signature** **Date**

Final approval and acceptance of the Thesis is contingent up on the submission of its final copy to the Council of Graduate Studies (CGS) through the candidate's Department or Faculty Graduate Committee (DGC or FGC).

BIOGRAPHICAL SKETCH

Mr. Mesele Gawo Emo, the author of this research paper, was born on June 21, 1986, E.C. in Biftu Town, Gurafarda Woreda, located in the South-West Ethiopia Region. He was born to his father, Mr. Gawo Emo Belku, and his mother, Mrs. Abebech W/Mariam.

He began his early education at Biftu Otowa Primary School and later continued his secondary education at Bebeka Secondary School. After completing his secondary studies, he joined Aman Polytechnic College, where he successfully completed a Diploma (Level IV) Training Program in the ICT sector, specializing in Hardware and Networking Services.

Driven by his strong interest in technology and education, Mesele pursued his higher education at Dilla University, where he earned a Bachelor of Science (B.Sc.) degree in Information and Communication Technology. Throughout his academic journey, he demonstrated dedication, discipline, and a strong passion for learning and professional growth.

In August 2011 E.C., he was employed at Aman Polytechnic College as an ICT Trainer. During his service at the college, he actively engaged in teaching, training, and supporting students in the field of Information and Communication Technology. Alongside his professional responsibilities, he continued advancing his education by joining the Master of Science (M.Sc.) program in Information and Communication Technology.

Currently, Mr. Mesele Gawo Emo is pursuing his graduate studies at the FDRE Technical and Vocational Training Institute. His research interest focuses on Natural Language Processing (NLP), particularly in the area of Amharic language processing using machine learning techniques.

As part of the fulfillment requirements for the Master of Science Degree in Information and Communication Technology, he is conducting and presenting his research study entitled “Amharic Text Chunking Using Machine Learning Approach.”

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to Almighty God, the Source of Wisdom and the Foundation of all knowledge, for giving me strength, guidance, and patience throughout my academic journey. I also extend my heartfelt thanks to the Holy Mother of the Lord, the Ever-Virgin St. Mary, for her continuous blessings, protection, and spiritual support during the successful completion of this study.

I would like to express my deepest appreciation and heartfelt thanks to the FDRE TVT Institute and the ICT Department for providing me with the opportunity, academic support, and learning environment necessary to accomplish this research successfully. My special appreciation and heartfelt thanks go to the postgraduate instructors: Brikinesh Woldeyohanes (Ph.D.), Ravindra Babu B.(Professor), Mohamed Tune(Ass. Professor), and Kibrom Tadese (Ph.D.) for their valuable lectures, professional guidance, encouragement, and the knowledge they shared throughout the course of my postgraduate studies. Their dedication greatly contributed to the success of this research work.

My special and heartfelt thanks also go to my advisor, Getnet Yilma (Ph.D.), for his patient supervision, constructive feedback, continuous encouragement, and unwavering support from the beginning to the completion of this study. His professional guidance, insightful comments, and commitment played a vital role in improving the quality of this research.

Furthermore, I would like to express my sincere gratitude and Thanks to my friends and colleagues for their encouragement and support during my studies. In particular, I would like to heartfelt thanks Zemedkun Solomon, Head and Curator of the Southwest Ethiopian People's Region Innovation and Industry Office. For his valuable encouragement and academic motivation. I am also deeply thankful to Mesfin Alemayehu Koins a Postgraduate Graduate of the FDRE TVT Institute ICT Department late (2012–2014 E.C.) and currently the Dean of Aman Polytechnic College, for his inspiring advice, ideas, and continuous support throughout my educational journey.

Finally, I would like to extend my deepest appreciation to my beloved family and fellow postgraduate classmates for their endless moral support, patience, encouragement, and understanding. Their kindness, motivation, and belief in me gave me the strength and confidence to overcome challenges and successfully complete my studies.

ACRONYM

ATC	Amharic Text Chunker
CRFs	Conditional Random Fields
HMM	Hidden Markov Model
MBL	Memory-Based Learning
MBT	Memory Based Tagger
ML	Machine Learning
MVDM	Modified Value Difference Matric
NLP	Natural Language Processing
NP	Noun Phrase
PP	Preposition Phrase
TC	Text Chunking
TiMBL	Tilburg Memory-Based Learner
VP	Verb Phrase
IOB	Inside Outside Beginning
POS	Part Of Speech
NER	Named Entity Recognition
WIC	Walta Information Center
AdjP	Adjective Phrases
IR	Information Retrieval
IE	Information Extraction
k-NN	k-Nearest Neighbors
MT	Machine Translation
NER	Named Entity Recognition
SVM	**Support Vector Machines**
MEMMs	Maximum Entropy Markov Models
FDRE	Federal Democratic Republic of Ethiopia
TVTI	Technical Vocational Training Institute

TABLE CONTENTS

DECLARATION i
APROVAL PAGE ii
BIOGRAPHICAL SKETCH iii
ACKNOWLEDGMENTS iv
ACRONYM v
LIST OF TABLES vi
ABSTRACT vi
CHAPTER ONE 10
1. INTRODUCTION 10
1.1 Background of Study 10
1.2 Statement of the Problem 11
1.3 Research Questions 12
1.4 The Objective of the Study 12
1.4.1 General Objective 12
1.4.2 Specific Objectives 12
1.5 Hypothesis of the Study 13
1.6 Scope of the Study 13
1.7 Limitation of the Study 13
1.8 Significance of the Study 14
1.9 Organization of the Research 15
CHAPTER TWO 16
2. LITERATURE REVIEW 16
2.1 Introduction 16
2.2 Rule Based Text Chunk Approach 16
2.3 Machine Learning Text Chunk Approach 17
2.3.1 Supervised Learning Approach 17
2.3.1.1 Support Vector Machines (SVM) 17
2.3.1.2 Hidden Markov Models (HMM) 18
2.3.1.3 Conditional Random Fields (CRFs) 18
2.3.1.4 Memory-Based Learning in NLP 20
2.3.2 Unsupervised Learning Approach 13
2.4 Hybrid Text Chunk Approach 13
2.5 Text Chunking Specification 14
2.6 Review of Related Works 14
2.6.1 Text Chunker for the Amharic Language 14
2.6.2 Text chunker for the Punjabi language 16

2.6.3 Text Chunker for Arabic Language 16
2.6.4 Text Chunker for Chinese Language 16
2.7 Research Gap 17
2.8 Summary 18
CHAPTER THREE 20
3 DESIGNS AND METHODOLOGY 20
3.1 Introduction 20
3.2 The Research Design Methodology 20
3.2.1 Research Design Method end to end Pipeline 20
3.2.2 Machine Learning Models for Chunking 22
3.3 Data Collection 22
3.4 Implementation Tools 22
3.4.1 Model Development Tools Used in the Study 23
3.5 Testing and Experimental Procedure 23
3.6 Basic Amharic Word Classes 24
3.6.1 Noun Class 24
3.6.2 Verb Class 24
3.6.3 Adjective Class 25
3.6.4 Adverb Class 25
3.6.5 Preposition Class 25
3.7 Amharic Phrasal Categories 25
3.7.1 Noun Phrase (NP) 25
3.7.2 Pronoun Phrase 25
3.7.3 Verb Phrase (VP) 26
3.7.4 Adverbial Phrase (AdvP) 26
3.7.5 Adjectival Phrase (AdjP) 26
3.7.6 Prepositional Phrase (PP) 26
3.7.7 Interjection Phrase 26
3.7.8 Determiner Phrase (DP) 26
3.7.9 Particle Phrase 26
3.8 Amharic Sentence Types 26
3.8.1 Simple Sentences 27
3.8.2 Compound Sentences 27
3.8.3 Complex Sentences 27
3.8.4 Compound-Complex Sentences 27
3.8.6 Negative Sentences 28
3.8.7 Interrogative Sentences 28
3.8.8 Imperative Sentences 28
3.8.9 Exclamatory Sentences 28

CHAPTER FOUR 31
4. AMHARIC TEXT CHUNKER MODEL 31
4.1. Introduction 31
4.1.1 Dataset Preparation 31
4.1.2. Overview of the Proposed System Design Model 32
4.2. Chunk Boundary Identification Using IOB2 Tagging Scheme 33
4.2.1. Description of chunk specification using IOB2 34
4.3 Proposed Amharic Text Chunk Model Architecture 34
4.3.1 Input Layer 34
4.3.2 Preprocessing Layer 35
4.3.3 Feature Extraction and Word Embedding Layer 35
4.3.4 Machine Learning Chunking Layer 36
4.3.5 Output Layer 36
4.4 Architecture of CRFs Based Amharic Text Chunk 37
4.4.1 CRF Training Process 38
4.4.2. CRF Testing Process 39
4.5 Architecture of MBL-based Amharic Text chunk 39
4.5.1. MB Training Phase 39
4.5.2. MBL Testing Phase 40
4.6 Applications of the ATC 41
4.7 Prototype of the ATC 43
CHAPTER FIVE 45
5. EXPERIMENT AND RESULT DISCUSSION 45
5.1 Introduction 45
5.2 Feature Representation and Description 45
5.3 Experiment in CRF-Based Amharic Text Chunking 45
5.3.1 CRF-Based Experiment 46
5.4 Memory-Based Text Chunk Implementation 48
5.4.1 Experiment with IB1 49
5.4.2 Experiments with IGTree 51
5.5 Dataset Distribution Analysis 54
5.5.1 Pre-SMOTE distribution of IOB2 chunk labels in ATC dataset 54
5.5.2 Balanced Class Distribution AFTER SMOTE in ATC Dataset 55
5.6 Baseline Study ATC Models 57
5.6.1 Hidden Markov Model (HMM) 58
5.6.2 Support Vector Machine (SVM) 58
5.6.3 Learning Based on Memory (IB1) 58
5.6.4 Memory-Based Learning (IGTree) 58
5.6.5 Reasons for Utilizing Baseline Models 59

5.7 Dataset Split/ Distribution 59
5.7 Proposed CRF Model Development 60
5.7.1 Chunk-level evaluation 60
5.7.2 Token-Level Evaluation 61
5.8 ROC Curve Evaluation of ATC Classification Performance 62
5.9 Measures of Model Performance 63
5.9.1 Model with CRF General Performance Indicators 63
5.9.2 IGTRee Model Performance Metrics in General 64
5.9.3 IB1 Model General Performance Indexes 65
5.9.4 SVM Model General Performance Metrics 66
5.10 Test Set Chunk Tag Distribution for Evaluating the ATC Model 67
5.11 Model Performance Analysis of the Study 68
5.11.1 General Model Performance Comparison Based on F1-score and Accuracy 68
5.11.2 F1-score and Accuracy Comparison of CRF, IB1 and IGTree Models 69
5.11.3 Performance of F1-Scores per Label across Models Compared 70
5.11.4 Model Precision, Recall, F1-score Accuracy 71
5.11.5 Comparing General Model Performance (CRF, IB1, IGTree, SVM) 72
CHAPTER SIX 73
6. CONCLUSION AND RECOMMENDATION 73
6.1 Conclusion 73
6.2 Recommendation 73
6.3 Future Work 74
REFERENCE 75
List of Appendix 79
Appendix 1: - Amharic ፊደል (Fidel or alphabet) Unicode standard 79
Appendix 2: - POS tags used by this study 80
Appendix 3: - Chunk Tag used by this study 81
Appendix 4: - POS-tagged and chunk annotated document for the training data set 82
Appendix 5: - Sample chunk tagged document for the training data set 83
Appendix 6. Data Collection Letter from Bench Sheko Community Television Service 84

LIST OF TABLES

Table 2.1 Research Gap 17
Table 3.1 Summary of tools and technologies used in this study 23
Table 4.1 Chunk Specification Using IOB2 34
Table 5.1 CRF-Based Experiment 47
Table 5.2 IB1-Based Experiment 50
Table 5.3 IGTRee Experiment 52
Table 5.4 distribution of training and testing set. 59
Table 5.5 Chunk-level evaluation (Weighted Average across all labels) 60
Table 5.6 Token-Level Evaluation 61
Table 5.7 CRF Model - Overall Metrics 64
Table 5.8 IGTRee Model - Overall Metrics 65
Table 5.9 IB1 Model - Overall Metrics 65
Table 5.10 SVM Model - Overall Metrics 66
Table 5.11 Model Performance Comparison Using F1-Score and Accuracy 68
Table 5.12 F1-score, Accuracy Comparison of CRF, IB1, IGTree Models 69
Table 5.13 Detailed F1-Scores per Label across Models 71
Table 5.14 Model Precision, Recall, F1-score Accuracy. 71
Table 5.15 Comparing General Model Performance (CRF, IB1, IGTree, SVM) 72
Table 1.1. Amharic ፊደል (Fidel or alphabet) Unicode standard 79
Table 2.2. POS tags used by this study 80
Table 3.3. Chunk Tag used by this study 81

LIST OF FIGURES

Figure 2.1 An Example of Memory based Architecture. 20
Figure 2.2. Example of Memory-Based Learning System. 12
Figure 3.1 Research design method end to end pipeline for Amharic text chunk 21
Figure 3. 2 Workflow diagram of proposed model 30
Figure 4. 1 Dataset used by the study and its description. 32
Figure 4.2. Proposed ATC Model Architecture Diagram 36
Figure 4. 3 Architecture of CRs based on Amharic text chunk. 37
Figure 4.4 Architecture of Memory-Based Learning of ATC. 41
Figure 4.5 Prototype of ATC 44
Figure 5.1 Class distribution before SMOTE (Token - Level). 54
Figure 5. 2 Chunk tag distribution before SMOTE (Token - Level) 55
Figure 5.3 Class distribution AFTER SMOTE (Token - Level) 56
Figure 5.4 Chunk tag distribution AFTER SMOTE (Token - Level) 57
Figure 5.5 distribution of training and testing set 59
Figure 5.6 Chunk Level Evaluation Metrics (Weighted Average) 60
Figure 5.7 Receiver Operating characteristic (ROC) 63
Figure 5.8 Model with CRF General Performance indicators 64
Figure 5. 9 IGTRee Model Overall Performance 65
Figure 5.10 IB1 Model Overall Performance 66
Figure 5.11 Chunk tag support counts in test set (from classification report) 68
Figure 5.12 Model Performance Comparison (F1-Score and Accuracy) 69
Figure 5.13 F1-score, Accuracy Comparison of CRF, IB1 and IGTree Models 70
Figure 5.14 Model Precision, Recall, F1-score Accuracy 71
Figure 5.15 Comparing General Model Performance (CRF, IB1, IGTree, SVM) 72
Figure 6.1. Data collection feedback letter 84

ABSTRACT

Amharic is one of the most widely spoken languages in Ethiopia and is commonly used in education, government offices, media and everyday communication. However, the development of Natural Language Processing (NLP) tools for Amharic is still limited mainly because of the lack of large, annotated datasets and effective language processing systems. One key NLP task is text chunking which means grouping words into meaningful parts such as noun phrases and verb phrases. This process is very important for many applications like machine translation, information retrieval,

Question answering and text summarization. In this study an Amharic text chunking system is developed using machine learning techniques to overcome the limitations of previous rule-based and statistical approaches. The main goal is to design and evaluate models that can automatically identify phrase structures in Amharic text focusing on Conditional Random Fields (CRF) and Memory-Based Learning (MBL) methods using different features and context window sizes.

To achieve this a supervised machine learning experimental design is used. A dataset of 12,000 Amharic sentences was manually annotated using the IOB2 tagging format, and the data are collected from various sources such as newspapers, grammar books, Wikipedia, BBC Amharic, government documents, and broadcasting media. The study implemented CRF and Memory-Based Learning models, including IB1 and IGTree, to test their performance in chunking tasks. The results showed that the CRF model performed better, achieving about 86% accuracy due to its strong ability to understand contextual and sequential relationships between words. In comparison, the MBL models performed slightly lower, with accuracy ranging from 79% to 85%, depending on the features used. Overall, the findings confirm that machine learning approaches, especially CRF are more effective and stable for Amharic text chunking, and the study suggests that future work should focus on larger datasets and advanced deep learning methods to further improve performance.

Keywords: *Amharic text chunker, machine learning, CRF, MBL, shallow parsing, IOB2 format, language technology, natural language processing*

CHAPTER ONE

1. INTRODUCTION

1.1 Background of Study

Today Natural Language Processing (NLP) plays a big role in our daily lives by helping computers understand human language (Mohammed Tigabu, 2023; Mulu & Goyal, 2020; TIGABU, 2023; Upgris & 2021, n.d.)One important task in NLP is text chunking (TC) which breaks down sentences into meaningful groups of words this helps with tasks like summarizing texts automatically or answering questions text chunking can be done as either shallow or deep parsing text chunking also called shallow parsing group's parts of a sentence into small chunks that belong together like noun phrases (NP), verb phrases (VP), prepositional phrases (PP), and adjective phrases (AdjP) the key idea is that each word belongs to one chunk only not multiple chunks for example in the sentence "*The great teacher read the new book*" (in Amharic: ታላቁ መምህሩ አዲሱን መፅሃፍ አነበበ) there are three main chunks:

- Noun Phrase (NP): *"The great teacher"*→ታላቁ መምህሩ
- Noun Phrase (NP): *"The new book"*→አዲሱን መፅሃፍ
- Verb Phrase (VP): *"Read"*→አነበበ

According to researchers (Mulu & Goyal, 2020; Upgris & 2021, n.d.) Text chunking is a useful alternative to full parsing which tries to analyze every detail of a sentence's structure but can slow down systems text chunking focuses on a higher level view that is usually enough for information retrieval (IR) and information extraction (IE) This means using chunking instead of full parsing can make IR and IE systems faster and more efficient many NLP and IR applications prefer chunking over full parsing for example to build a good question-answering system a chunk is essential (Mohammed Tigabu, 2023; Mulu & Goyal, 2020; Upgris & 2021, n.d.) Finding noun and verb phrase is often enough for IR systems text chunking can also be used in text mining, automatic summarization.

Researchers have studied text chunking in many languages worldwide including Arabic (Abolohom et al., 2021; Alnagi et al., 2025a, 2025b) Chinese (Chen et al., n.d.-b; Y. Zhao et al., n.d.) Bengali (Sarkar et al., n.d.)Amharic (Mulu & Goyal, 2020; Yimam et al., 2021), Urdu (Zain et al., 2025b) so far there are various study has focused on amharic from those done by (Upgris & 2021, n.d.) however this study did not cover all types of amharic sentences it used Hidden Markov Models (HMM) for chunking and error correcting rules but HMM has known problems like label bias Because Amharic is a morphologically rich and highly inflected language with unique letters and grammar existing chunker from other languages cannot be directly applied to it.(Molina et al., n.d.)

Text chunking can be done mainly in two ways rule based or machine learning approaches Rule based systems rely on many handcrafted rules which require deep linguistic knowledge and are time consuming to build they also don't easily adapt to other languages.

Machine learning on the other hand learns rules automatically from labeled data. It has two main types supervised and unsupervised learning. Supervised learning uses labeled datasets to train the model while unsupervised learning does not. however unsupervised methods often treat features independently and struggle to capture relationships leading to lower accuracy This is why supervised learning is preferred for Amharic text chunking one popular supervised method is Conditional Random Fields (CRFs) which are great for sequential data tasks like text chunking, named entity recognition, and part-of-speech tagging (Mulu & Goyal, 2020; Upgris & 2021, n.d.) And another method is Memory Based Learning (MBL) which solves new problems by comparing them to previously learned examples using similarity measures like k-Nearest Neighbors (k-NN) MBL has shown promising results in text chunking for other languages.(Fake et al., 2024; Mohammed Tigabu, 2023)

This research focus to explore and build an amharic text chunker using supervised machine learning techniques specifically CRFs and MBL and manually tagged our training data with part-of-speech and chunk tags using IOB1 format then trained both CRF and MBL models to find best way to chunk amharic sentences accurately, efficiently and lastly assessed the models' performance using test dataset.

1.2 Statement of the Problem

Amharic is one of the most widely spoken languages in Ethiopia and the second most widely spoken Semitic language after Arabic. Despite its large number of speakers, the language still has limited research and computational resources in the field of Natural Language Processing (NLP). One of the major missing components is an effective text chunking system. Text chunking is an important NLP task that divides sentences into meaningful phrases such as noun phrases and verb phrases, which helps computers better understand sentence structure and meaning.

- Text chunking is widely used in many NLP applications such as machine translation, information extraction and information retrieval, question answering systems, chatbots and speech processing systems. (TIGABU, 2023; Upgris & 2021, n.d.) Without an accurate chunking system, the performance of these applications becomes limited because computers may fail to correctly identify relationships between words and phrases in a sentence.
- Although many languages have developed advanced chunking systems using modern machine learning approaches, Amharic has received very little attention in this area. The main previous work on Amharic text chunking used a Hidden Markov Model (HMM). However, the system faced several challenges, including limited annotated training data, low prediction accuracy, label

bias problems, and difficulty handling complex sentence structures. (Ibrahim & Assabie, 2016) As a result, the existing system was not sufficiently effective for advanced NLP applications.

- In contrast, other languages such as Arabic, Chinese and Punjabi have improved text chunking performance by using modern machine learning methods such as Conditional Random Fields (CRF) and Memory-Based Learning (MBL).(Gupta & Gupta, 2022) (Abolohom et al., 2021; Chen et al., n.d.-b)These approaches are better at capturing contextual information and handling complex language patterns. However, Amharic and many other Ethiopian languages have not fully benefited from these modern techniques due to the lack of annotated datasets and limited research efforts.
- The main problem addressed in this study is the absence of a reliable and accurate Amharic text chunking system based on modern machine learning approaches. This study aims to solve this problem by developing an improved Amharic text chunker using CRF and MBL models together with rich linguistic and contextual features. The research also aims to improve chunk prediction accuracy and support the development of advanced NLP applications for the Amharic language.

1.3 Research Questions

1. Which linguistic features such as part-of-speech tags, word embeddings, or morphological patterns play the most important role in improving chunking accuracy?
2. How can a machine learning system be designed to effectively identify meaningful chunks in Amharic sentences using a variety of linguistic features?
3. How does the performance of chunking models differ when applied to different types of Amharic text such as formal writing from newspapers versus informal language from social media?

1.4 The Objective of the Study

This study focuses on developing a machine teaching based on Amharic chunker that identifies noun phrases, verb phrases and other sentence components to enhance translation, speech recognition and language understanding tools.

1.4.1 General Objective

The General Objective of this study is to investigate text chunking for Amharic using Conditional Random Fields (CRF) and Memory Based Learning (MBL).

1.4.2 Specific Objectives

In order to achieve the main objective and answers to the research questions the following specific tasks are done.

A. To study the basic syntactic structure of Amharic sentences (i.e., phrase structure, word category, sentence construction etc.).

B. To design a generic model for MBL and CRFs based text chunker for Amharic.

C. To develop a prototype of the chunker that accepts Amharic text as an input and produces chunked text as an output.

D. To evaluate the proposed methods performance for Amharic text chunking.

1.5 Hypothesis of the Study

This study is based on the belief that building an effective Amharic text chunker using machine learning depends on choosing the right language features and understanding how different types of Amharic text behave. The following points summarize the main assumptions:

- Using key linguistic features like part-of-speech tags, text forms and context-based text meanings will help the chunking system perform more accurately.
- A machine learning model that is carefully designed and trained on real Amharic sentences will be able to correctly identify useful chunks of text such as noun and verb phrases.
- The system is expected to work better on formal Amharic texts like newspapers where grammar and structure are more consistent and may face more challenges with informal texts like social media which often include slang, typos or mixed languages.
- These ideas guide the system's design and help shape how the model is tested and improved throughout the study

1.6 Scope of the Study

This study aims to build a system that can automatically group text in Amharic sentences into meaningful chunks like noun phrases or verb phrases using machine learning. The goal is to make it easier for computers to understand Amharic text.

The research focuses only on Amharic using text from Ethiopian news, broadcast media and social Governmental publication so the system can learn from both formal and informal writing to make the chunker effective the study uses important features like part-of-speech tags, text structures and context-based text meanings and different machine learning models such as CRF and BiLSTM are tested to find out which works best for the system focuses on chunking only not full sentence analysis and it does not include other Ethiopian languages overall the study is meant to support the development of Amharic language tools that can be useful in real-life applications like translation, search engines and educational software.

1.7 Limitation of the Study

This study has several limitations and challenges that should be clearly acknowledged as they affect the system’s overall performance and future scalability.

- One of the main limitations is related to the dataset (Mohammed Tigabu, 2023; TIGABU, 2023)The study uses a manually annotated corpus of about 12,000 Amharic sentences. Although this is a valuable contribution compared to previous work, the dataset size is still relatively small

for machine learning models. As a result, the system may not fully capture all grammatical patterns of the Amharic language, and its performance on unseen or real-world data may be limited. In addition, there is a general lack of publicly available Amharic NLP resources, such as large corpora, standardized annotation schemes, and open-source tools. Another issue is that the dataset mainly consists of formal text sources like newspapers, government documents, and broadcast materials. This means the model may struggle when processing informal language such as social media text or spoken conversations.

- Another important challenge comes from the linguistic nature of Amharic itself. Amharic is a morphologically rich language meaning that words can change form in many complex ways depending on grammar and context. This makes it difficult for the model to fully capture all word-level and sub-word-level patterns. In some cases, important parts of a phrase may be spread across multiple words in a sentence, requiring the system to understand a wider context to correctly identify chunks.
- There are also model-related limitations The studies focuses mainly on traditional machine learning methods, specifically Conditional Random Fields (CRF) and Memory-Based Learning (MBL). In addition, some configurations of the Memory-Based Learning model are sensitive to small changes in input formatting or noisy features, which can negatively affect performance. The system is also limited to shallow parsing, meaning it only groups words into basic phrases (like noun phrases and verb phrases) rather than performing a full deep syntactic analysis of sentences.(4715182, n.d.; Shiferaw et al., 2024)
- There are limitations inherited from previous baseline systems**.** Earlier approaches, especially those based on Hidden Markov Models (HMM), suffer from issues such as label bias, low prediction accuracy and difficulty handling long or complex sentence structures. These weaknesses highlight the need for more advanced methods in Amharic text chunking. (Upgris & 2021, n.d.)

1.8 Significance of the Study

Text chunking is an important task in Natural Language Processing (NLP) because it divides sentences into meaningful units such as noun phrases and verb phrases helping computers better understand language structure. It improves the performance of applications such as information extraction, named entity recognition (NER), machine translation, information retrieval, question answering, chatbots, language sentiment analysis, speech recognition and text summarization (Upgris & 2021, n.d.; Yimam et al., 2021)This study is significant because it contributes to the development of NLP resources and computational tools for the Amharic language which is still a low-resource language in NLP research. The study develops an automatic Amharic text chunking system using Conditional Random Fields

(CRF) and Memory-Based Learning (MBL). It also creates a manually annotated dataset approximately 12,000 Amharic sentences labeled with POS tags and IOB2 chunk tags.

- And the study compares machine learning models such as CRF, IB1, IGTree, and SVM to identify effective approaches for Amharic text chunking. By integrating contextual and linguistic features such as POS tags, neighboring words, morphological patterns, and word embeddings, the proposed system improves chunk prediction performance and sentence understanding.
- Overall, the developed chunking system and annotated dataset provide a foundation for future Amharic NLP applications, including machine translation, chatbots, search engines, speech recognition, and intelligent information retrieval systems.

1.9 Organization of the Research

This thesis study is organized into six chapters the next chapter. Chapter two provides a review of the relevant literature. It covers different approaches to text chunking, its specifications and previous related work. Chapter three explores key aspects of the Amharic writing system that are relevant to this study focusing on how Amharic text can be represented and processed electronically. In Chapter Four the process of preparing the dataset and building the model is explained. Chapter Five presents the experimental set-up details of the steps taken during experimentation and discusses the results. Finally, Chapter Six concludes the thesis with a summary of key findings and offers recommendations based on the outcomes of the research.

CHAPTER TWO

2. LITERATURE REVIEW

2.1 Introduction

This chapter explains an overview of how the Amharic text chunking system will be developed using modern Natural Language Processing (NLP) techniques the methods are carefully selected to address the research questions and meet the project's goals from collecting relevant Amharic data to producing accurate linguistic outputs such as noun and verb phrase chunks.

To set the stage the chapter primarily introduces the broader concept of text chunking a key task in (NLP) (Hwang et al., n.d.; J. Wang et al., 2024; Z. Wang et al., 2024) that involves dividing sentences into meaningful units like Noun Phrases and Verb Phrases these chunks are essential for advanced applications such as syntactic parsing, information extraction and machine translation areas increasingly important in Ethiopia's growing digital landscape and ecosystem and the chapter also reviews key research in the field especially the shift from early rule-based systems to more advanced Machine Learning and deep learning models while major languages like Chinese and Arabic have seen rapid progress and low-resource languages like Amharic remain underexplored it is rich morphology, unique script, and lack of annotated data pose major challenges so far only one rule-based Amharic chunker has been attempted, underlining the need for more robust and scalable solutions something this study aims to deliver and Text chunking is an important area of study within computational linguistics focusing on the task of dividing text into syntactically meaningful groups such as noun or verb phrases researchers have explored various approaches to implement chunking systems effectively and the three main categories of approaches are rule based text chunk, Machine learning text chunk and hybrid text chunk. (Balek et al., 2025a, 2025b; Sietsma et al., 2024)

2.2 Rule Based Text Chunk Approach

The rule - based approach to text chunking known as the knowledge, based or linguistics-based method is considered one of the more straightforward techniques it relies on manually crafted linguistic rules that define how to identify chunks within a text developing such a system requires a deep understanding of the target language's grammar and structure. And methods which rely on handcrafted linguistic rules. One of the key advantages of the rule-based approach is its simplicity and the fact that it does not require any training data however it also comes with several limitations creating an effective rule-based system demands extensive linguistic expertise and the development of a large number of rules to account for all the language's features additionally these systems are highly language specific making them difficult to adapt to other languages the process is time consuming and often challenging to scale or extend once implemented.

2.3 Machine Learning Text Chunk Approach

Text chunking is an important area of study within computational linguistics focusing on the task of dividing text into syntactically meaningful groups such as noun or verb phrases researchers have explored various approaches to implement chunking systems effectively and the three main categories of approaches are machine learning techniques which use data driven models to learn patterns from annotated corpora. (Balek et al., 2025a, 2025b; Sietsma et al., 2024)

The machine learning approach to text chunking offers a more flexible and scalable alternative to traditional rule based methods instead of relying on hand-crafted linguistic rules the system automatically learns patterns from a dataset this means developers do not need deep linguistic expertise to build a chunker the model learns the structure from examples one of the main advantages of this approach is its ability to adapt to different languages or domains quickly provided there is enough annotated data however the quality and size of the training dataset are critical the better data, more accurate the model's predictions. (Dahia & Belbacha, 2024; Qi & Shabrina, 2023; Xu et al., 2024)

2.3.1 Supervised Learning Approach

Supervised learning automatically generates chunking rules from annotated training data. This means the developers do not need deep domain knowledge since the system learns from examples however the training data must be carefully labeled typically with information like part-of-speech (POS) tags and chunk tags that reveal the structure of the text**.**

Supervised learning uses probabilistic and statistical methods to analyze the syntactic structure of language common machine learning techniques used in this approach include Support Vector Machines (SVM) (AlBadani et al., 2022a)**,** Hidden Markov Models (HMM) (Molina et al., n.d.)Conditional Random Fields (CRFs) (Suraksha et al., n.d.; Upgris & 2021, n.d.), Memory-Based Learning (MBL) (Kanwal et al., 2023; Mishra et al., 2023)Each method has its strengths and the choice of which to use often depends on the specific problem and the language being processed.

In this study chose to use Conditional Random Fields (CRFs) and Memory-Based Learning (MBL) for developing a text chunker for Amharic this decision was partly driven by the limitations of earlier attempts that relied on a hybrid approach combining HMM with rule-based methods CRFs & MBL have shown strong performance in various NLP tasks such as Named Entity Recognition (NER), POS tagging, Semantic labeling, Morphological analysis and Text chunking because of their versatility and proven success in language processing were motivated to explore their use in building a more effective chunker for the amharic language.

2.3.1.1 Support Vector Machines (SVM)

Support Vector Machines (SVM) is a machine learning method that learns how to separate and organize text by drawing clear boundaries between different language patterns when used for a text

chunker, SVM looks at each word in a sentence along with its surrounding words, part-of-speech tags and other linguistic features and decides whether that word belongs to the beginning of a chunk, inside a chunk or outside a chunk. (AlBadani et al., 2022b; Hossain, 2022)

- It does this by finding the best possible boundary (or "decision line") that separates different chunk types based on the training data.
- SVM acts as a careful judge. After studying many correctly chunked sentences it learns where the boundary between phrases usually appears when it sees a new sentence, it uses what it has learned to place chunk boundaries as accurately as possible, even for sentences it has never seen before

2.3.1.2 Hidden Markov Models (HMM)

HMM are machine learning models that learn text patterns by looking at sequences of words and guessing the hidden structure behind them (Models & Haugh, n.d.)HMM treats a sentence as a sequence of words and assumes that each word belongs to a hidden chunk label such as beginning of a phrase, inside a phrase or outside a phrase. These chunk labels are not directly visible but the model learns to predict them by observing how words and their features usually appear in order.

- HMM works like a reader who follows the flow of a sentence. It not only looks at the current word but also considers what came before it and what is likely to come next by learning common word sequences and phrase patterns from training data, the model becomes good at predicting where one chunk ends and another begins.

2.3.1.3 Conditional Random Fields (CRFs)

CRFs are a type of probabilistic machine learning model used for labeling sequences of data. In natural language processing (NLP), they are widely used for tasks such as part-of-speech tagging, named entity recognition, and text chunking, each word in a sentence must be assigned a meaningful label based on its surrounding context unlike approaches that treat each word independently CRFs consider the entire sentence as a connected sequence. This means the prediction of each word depends not only on the word itself but also on the previous and neighboring words. Given an input sentence represented as a sequence of words: X=(X1,X2…,Xn) and its corresponding output label sequence: Y=(Y1,Y2…,Yn) the main objective of the CRF model is to find the most probable label sequence Y given the input X. To model this relationship, a linear-chain CRF defines the conditional probability of a label sequence as(Mulu & Goyal, 2020; Upgris & 2021, n.d.)As described above CRFs are probabilistic model it computes the probability of P (X|Y) of possible output in equation 1.

$$\boldsymbol{P(Y|X) = \frac{1}{Z(X)} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda i\, fi(Y_{j-1}, Y_j, X, j)\right)} \qquad \boldsymbol{eq\ 1}$$

In text chunking, a sentence is treated as a sequence of words X=(X1, X2…, Xn) and each word is assigned a corresponding label sequence Y= (Y1, Y2…, Yn). Here, X represents the input words, while

models this problem by predicting the most likely label sequence based on the entire sentence. Instead of labeling each word independently, it considers the relationship between neighboring labels and the context of the words.

In simple terms, this equation shows that the probability of a label sequence is determined by combining many small pieces of information (called features) extracted from the sentence. Each feature contributes to the final decision, and each one has a learned importance weight.

Here is $\boldsymbol{fi(Y_{j-1},Y_j,X,j)}$ is a feature function that captures useful information at position of **j** such as the current word or its context and **j** indicate the position of input sequence, $\boldsymbol{\lambda_i}$ (lambda) is the weight assigned to each feature learned during training to show how important that feature is, $\boldsymbol{n}$ represents the length of the sentence, $\boldsymbol{m}$ represents the number of feature functions or template used, to ensure that Assigned to the different features in the training phase, $Z_{\lambda^{\rightarrow}}(X)$ is normalization factor that make the probability in the range [0,1], which can be expressed as (Abeba, 2013; Mohammed Tigabu, 2023)CRFs use a normalization factor called the partition function see the equation 2 below.

$$\boldsymbol{Z(X) = \sum_{Y'} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i \, fi\left(Y'_{j-1}, Y'_j, X, j\right)\right)} \qquad \boldsymbol{eq\ 2}$$

Feature functions are the core building blocks of CRFs. A feature function: $\boldsymbol{fi\ (Y_{j-1},\ Yj,\ X,\ j)}$ represents a simple condition or pattern observed in the data at position **j**. For example, it may check whether: the current word is a noun or verb, the word has a specific form or structure or the word belongs to a certain chunk type. In this study three main feature types are used they are the actual word (token), its Part-of-Speech (POS) tag and its IOB chunk tag, which indicates whether a word is Inside, Outside, or at the Beginning of a phrase.

These features can also combine to create richer contextual information example the model may learn patterns such as a capitalized word followed by a verb being more likely to represent named entity. Each feature is assigned a weight $\boldsymbol{\lambda_i}$ (lambda) which is learned during training. Features that help improve prediction accuracy receive higher weights while less useful features receive lower weights.

In simple terms, a CRF works by considering all possible label sequences for a sentence and assigning each one a score. The model then selects the sequence with the highest score as the final prediction.

This decision is based on two key ideas those are the relationship between neighboring labels, such as $\mathbf{Y_{j-1}}$and $\mathbf{Y_j}$, and the contextual information from the input sentence at each position.

Because CRFs evaluate the whole sentence as a structured unit rather than individual words, they are highly effective for sequence labeling tasks. This makes them particularly suitable for applications such as Amharic text chunking, where understanding context is essential for accurate labeling.

2.3.1.4 Memory-Based Learning in NLP

Memory-based language processing (MBLP) is a machine learning approach that works by directly reusing examples to solve natural language processing (NLP) problems it has two main parts the learning component and the performance or processing component during training the learning part Simply stores examples in memory without abstracting, selecting or reorganizing them this is why it's often called "lazy learning."(Mogridge & Ragni, 2024)

Memory-based learning also known as example-based learning, lazy learning or instance-based learning is a symbolic machine learning approach that blends elements of both probabilistic and knowledge based methods it is especially useful in natural language processing (NLP) tasks which often involve large datasets and many features of varying importance (Kanwal et al., 2023; Mishra et al., 2023; Mogridge & Ragni, 2024; Rekha Raj & Reghu Raj, 2016)The basic idea is simple when the system is presented with a new piece of data it does not try to generalize from past experiences right away like many traditional machine learning models do

Instead, it stores all previous examples and compares the new input to them assigning a label based on the closest match this process relies on a distance metric denoted as Δ(X; Y) which measures the

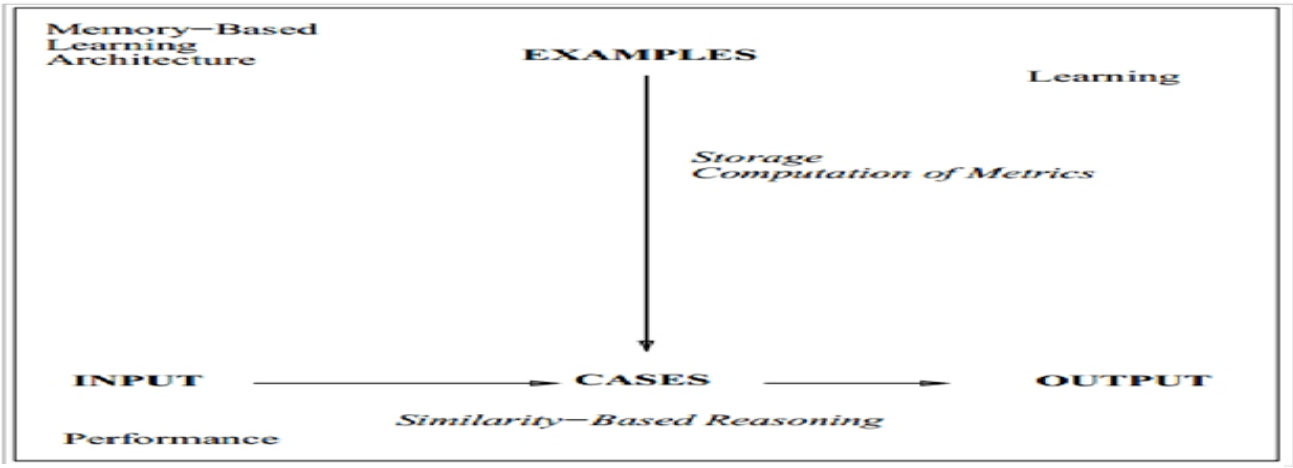


Figure 2.1 An Example of Memory based Architecture.

similarity between the new input (X) and each stored example (Y) to efficiently handle large amounts of data memory based learning systems use advanced data structures and acceleration techniques inspired by the k-nearest neighbors (k-NN) algorithm this allows the system to scale well even in complex scenarios with vast feature spaces as introduced in Chapter One the most widely used tools for memory based learning is TiMBL (Tilburg Memory-Based Learner) a free and open source software package TiMBL was the University of

Antwerp it is versatile and can be used in different ways through a command line interface as a server or integrated into other programming environments like Python via a C++ API.

Figure 2.2 shows how a memory-based learning system functions the encoder (γ) transforms an input from the input space (X) into one or more memory addresses and the decoder (β) maps the select

Memory entries to an output in the output space (Y) the system is memory structure can vary it might use hash tables tree-based structures or full search tables to organize stored examples.

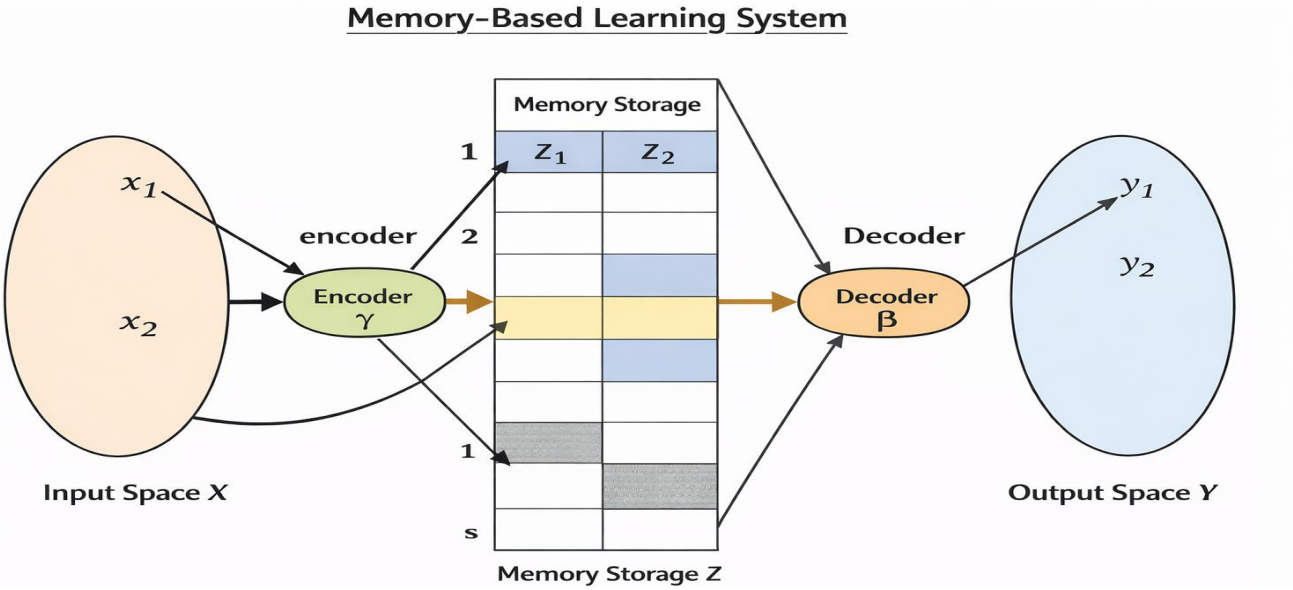


Figure 2.2. Example of Memory-Based Learning System.

As mentioned earlier the Tilburg Memory Based Learner (TiMBL) is a software package that implements several memory-based algorithms using different similarity measures some of the algorithms included in TiMBL are IGTree, IB1, IB1-IG, TRIBL, TRIBL2, IB2, and C4.5 the similarity measures it uses include Overlap, Modified Value Difference, Jeffrey Divergence, Do product and Cosine. Alongside these algorithms and similarity measures TiMBL also offer

Feature weighting methods such as Gain Ratio, Information Gain, Chi-squared and Shared Variance. All of these algorithms share a common approach they explicitly store a representation of the training data in memory when testing new instances are classified by comparing them to the most similar examples stored the main differences between the algorithms lie in how similarity is defined, how instances are stored and how the search through memory is conducted in this research to implement the IB1 and IGTree algorithms using the Overlap and Modified Value Difference similarity measures. Memory based learning has several advantages over other machine learning methods first it supports incremental training with little to no extra cost new data can be added and immediately influences the model without needing to retrain from scratch, second unlike many machine learning approaches that forget old training examples memory based learning retains all training data finally it is language independent making it suitable for a wide variety of languages.(Mogridge & Ragni, 2024; Report, 2018)

This feature motivated to explore memory-based learning for developing an Amharic text chunker. The main drawback of memory-based learning is its slower classification speed compared to other methods in the worst case the classification time is proportional to **O (nf)** where **n** is the number of

stored examples and **f** is the number of features this is because each new example must be compared to all stored examples across all features.

2.3.2 Unsupervised Learning Approach

Unsupervised Learning systems reduce the burden of the user which spent in annotating training data it requires only a statement of the required information no chunking patterns are given in advance by the user in this learning an annotated corpus is not used to improve the system's level of performance. The main challenge is to realize the user's need into a set of the chunking patterns the systems are based on bootstrapping methods expanding an initial small set of chunking patterns like rule based approach machine learning also has its own advantage and disadvantage the advantage of machine learning approach is that it can be transferred to a different domain easily as long as specific texts and a person who can annotate them are available but sometimes those texts are problematic or expensive to obtain or there is a lack of useful Click or tap here to enter text.

Documents on which an algorithm can learn and manual (even machine aided) annotation on the scale needed to provide reasonable levels of performance may be expensive.

2.4 Hybrid Text Chunk Approach

Text chunking is an important area of study within computational linguistics focusing on the task of dividing text into syntactically meaningful groups such as noun or verb phrases researchers have explored various approaches to implement chunking systems effectively and the three main categories of approaches are machine learning techniques which use data driven models to learn patterns from annotated corpora. Hybrid approaches which combine the strengths of both rule based and machine learning methods to improve performance and adaptability (Balek et al., 2025a, 2025b; Sietsma et al., 2024) These diverse strategies have enabled the development of chunking systems for a wide range of languages and applications.

A Hybrid Approach in text chunking is a method that combines two or more techniques usually different machine learning models or machine learning with linguistic rules to achieve better results than using a single approach alone.

In a text chunker a hybrid system may use one model (such as HMM or CRF) to learn the sequence and flow of words while another method (such as SVM, Memory-Based Learning, or hand-crafted linguistic rules) helps refine decisions about chunk boundaries each method contributes its strengths and compensates for the weaknesses of the others.

a hybrid approach works like a team instead of relying on one “expert,” the system listens to multiple experts and combines their opinions to decide where phrases begin and end. This leads to more accurate and robust chunking.(Alnagi et al., 2025a; El Mir et al., 2022)

In the hybrid approach both machine learning and rule-based methods are combined allowing the system to benefit from the strengths of each there are different ways to implement this Approach depending on the task in some cases a rule-based system is applied first and then machine learning is used to handle exceptional cases that the rules don't cover for example in paper a hybrid method was used for Korean text chunking: the authors first applied a rule based approach and then used a memory based machine learning technique to manage exceptions to the rules alternatively the process can start with machine learning followed by rule based error correction for cases where the machine learning model makes mistakes for instance paper employed this method for amharic base phrase chunking they initially used an HMM model for chunking and then applied error pruning rules to fix the chunks that the HMM incorrectly labeled. (Alnagi et al., 2025a; El Mir et al., 2022; Zain et al., 2025b, 2025a)

2.5 Text Chunking Specification

There are various types of chunk tags and techniques for identifying chunk boundaries chunk tags can represent different phrase types such as noun phrases, verb phrases, adverb phrases and so on depending on the syntactic rules of the target natural language to determine the boundaries of each chunk within a sentence seven boundary tagging formats are commonly used IOB1, IOB2, IOE1, IOE2, IO, "[" and "]" the first four IOB1, IOB2, IOE1, and IOE2 are complete chunk representation formats these formats mark the beginning and end of phrases and use. .

- **I** (Inside): for tokens that are inside a phrase.
- **O** (Outside): for tokens that are outside any phrase.

Where they differ is in how they mark the beginning or end of chunks:

- **IOB1** a token at the start of a chunk that directly follows another chunk is tagged as **B** (Beginning).
- **IOB2**: All chunk-initial tokens are marked with a **B** tag, regardless of context.
- **IOE1:** A token at the end of a chunk that is immediately followed by another chunk of the same type is tagged as **E** (End).
- **IOE2**: All chunk-final tokens are tagged with **E.**

The remaining three formats provide partial chunk representations.

- **IO**: Tokens inside a chunk receive **I** tag all others are tagged with **O.**
- **[**: Marks all chunk-initial tokens with **[,** and all other tokens with a default tag.
- **]**: Marks all chunk-final tokens with], while other tokens receive a default tag.
- These tagging schemes help systems identify the structure and organization of phrases within a sentence for tasks such as parsing, information extraction and machine translation.

2.6 Review of Related Works

As mentioned earlier text chunking has been developed for many languages around the world using a variety of approaches in this section will explore the different tools, techniques and experimental results presented by various scholars in the field of text chunking.

Researchers have applied both traditional machine learning methods and modern deep learning techniques to tackle the challenges posed by different languages these studies offer valuable insights into how linguistic features and model architectures affect chunking performance by reviewing their findings aim to understand which methods work best and identify gaps that future research can address text chunking breaking down sentences into meaningful syntactic units like noun phrases and verb phrases is essential for many NLP applications including parsing and machine translation. (El Mir et al., 2022; Gunasekaran, n.d.)

Although research on Amharic text chunking is still emerging most studies have focused on supervised machine learning methods trained on annotated corpora early efforts primarily relied on rule based and statistical approaches that used hand crafted linguistic rules grounded in Amharic grammar while these methods showed promise in certain cases they lacked flexibility and often struggled to handle the language's diverse and complex structures in the following sections review the main tools, methods, and findings from previous studies on Amharic text chunking discussing their strengths and limitations.

2.6.1 Text Chunker for the Amharic Language

As mentioned in Chapter One various major study has been conducted on Amharic text chunking carried out by the main goal of that research was to develop a base phrase chunker for the Amharic language and to extend it into a full syntactic parser the researchers employed a Hidden Markov Model (HMM) (Y. R. Zhao & Chen, 2024; Zhou, n.d.)to build the initial chunker to improve accuracy they also designed additional rules to correct chunking errors made by the HMM for converting the chunker into a full parser a bottom up approach was used along with a transformation based algorithm to identify phrase boundaries the Inside-Outside-Beginning (IOB) tagging scheme was applied.

For training and testing 320 diverse Amharic sentences were collected from grammar books and news articles from the News Walta Information Center (WIC). Of these 288 sentences were used for training and the remaining thirty-two for testing. Since the grammar book sentences were not pre-tagged both POS tagging and chunk tagging were performed manually one of the main challenges encountered during the study was tag sequence conflict which the researchers highlighted as a significant source of errors they recommended further investigation into this issue for future research. Additionally, the small size of the dataset limited the system's performance furthermore as noted (Y. R. Zhao & Chen, 2024) The HMM model used in the study suffers from a label bias problem while the researchers

implemented error pruning rules to fix some of the incorrect chunking the more serious errors particularly those related to tag sequence conflicts remained unresolved.

2.6.2 Text chunker for the Punjabi language

As (Sharma et al., n.d.) Summary of Hidden Markov Model (HMM**)** this study introduces a chunking system for the Punjabi language using a Hidden Markov Model (HMM) the system uses the Viterbi algorithm to determine the most likely chunk sequences and is trained using the Baum Welch algorithm with 25,000 lines of chunked Punjabi text this paper presents a chunking approach for the Punjabi language using Hidden Markov Models (HMM) a statistical method widely used in Natural Language Processing (NLP) chunking refers to the process of breaking text into syntactically related word groups called "chunks" and assigning appropriate labels to them the chunker was developed by applying the Viterbi algorithm to identify the most probable chunk sequences and the system was trained using the Baum Welch algorithm on a dataset of 25,000 lines of pre chunked Punjabi text for evaluation a separate annotated dataset of 1,000 lines was used to test the model's performance the Results is the system achieved approximately 80% accuracy in detecting chunk boundaries labeling accuracy assigning correct chunk labels was about 82% this work shows promising results and highlights the potential of HMM based methods for chunking in low resource languages like Punjabi.

2.6.3 Text Chunker for Arabic Language

(Khoufi et al., n.d.) discussed chunking Arabic text using Conditional Random Fields (CRFs) for their experiments they used the Penn Arabic Treebank as the dataset the features included two preceding and two following words relative to the current word as well as the corresponding POS tags for those surrounding words they also incorporated IOB chunk boundary identification the researchers used a total of 12,624 sentences with 10,100 sentences (80%) for training and the remaining 2,524 sentences (20%) for testing their model achieved an overall accuracy of 83.54%.

2.6.4 Text Chunker for Chinese Language

(Chen et al., n.d.-a) Aimed to improve Chinese functional chunk parsing by incorporating clause boundary information they first developed a rule-based method to automatically identify clause boundaries using punctuation and predicate information this clause boundary data was added as a new feature to their Conditional Random Fields (CRFs) model in addition to clause boundaries they also included POS tags as features the results of their experiments showed that incorporating clause boundary information significantly enhanced the performance of Chinese functional chunk parsing particularly by addressing the challenge of parsing long functional chunks.

2.7 Research Gap

NLP has advanced significantly for many languages Amharic is still thought of as a low-resource language with little syntactic processing development particularly in text chunking. Earlier research mostly used conventional techniques like rule-based methods and Hidden Markov Models (HMM) which usually have low correctness, label bias and difficulty with long and difficult sentence constructions in Amharic. Conversely Conditional Random Fields (CRF) and Memory-Based Learning (MBL) two more sophisticated models that have proven great performance in Arabic, Chinese and Punjabi have not been extensively studied for Amharic. Because Amharic is a morphologically rich language lack of rich contextual feature integration is still another significant barrier. Furthermore impeding adequate model training and assessment is a lack of big consistent annotated datasets in IOB2 format. Furthermore absent are thorough comparative studies of several Amharic chunking machine learning methods. This study creates a 12,000-sentence IOB2 dataset and constructs an Amharic chunking system using CRF, IB1, IGTree and SVM models to solve these problems enable fair comparison and advance study in Amharic NLP.

Table 2.1 Research Gap

Author(s)	Year	**Method / Focus**	**Key Findings**	**Research Gap**
Abeba	2020	Hidden Markov Model (HMM) with rule-based Amharic chunking	Developed an early Amharic chunker using HMM and error-correction rules. Worked on a small dataset using IOB tagging format.	Suffered from low accuracy, label bias issues, and difficulty handling complex Amharic sentence structures.
Ibrahim & Assabie	2021	Improved HMM-based chunking	Addressed tag sequence inconsistencies and slightly improved rule-based performance.	Still limited by HMM weaknesses; no adoption of stronger models like CRF or memory-based learning.
Mulu & Goyal	2020	General Amharic NLP chunking study	Emphasized the importance of supervised learning and contextual feature engineering.	Lack of large-scale annotated datasets and weak integration of linguistic features.
Upgris	2021	Rule-based and statistical chunking methods	Demonstrated baseline chunking performance using traditional approaches.	Methods do not scale well for modern or complex NLP applications.

Mittal et al.,	2020	HMM-based chunking for Punjabi language	Achieved around 80% accuracy using a large dataset.	Shows HMM can work for other languages, but it is not optimal for structurally complex languages like Amharic.
Khoufi et al.	n.d.	CRF-based Arabic chunking	Achieved 83.54% accuracy using rich POS and contextual features.	No comparable CRF-based benchmark systems available for Amharic.
Chen et al.	n.d.	CRF with clause boundary features for Chinese chunking	Improved chunking accuracy using clause boundary and contextual linguistic features.	Advanced contextual and clause-level feature integration has not been fully explored for Amharic NLP research.
Mohammed Tigabu	2023	Amharic NLP resource and chunking challenges	Highlighted the shortage of Amharic NLP resources, annotated corpora, and computational tools.	Lack of publicly available large-scale standardized datasets for Amharic chunking research.
Upgris	2021	CRF with clause boundary features for Chinese	Improved chunking accuracy using advanced contextual and clause-level features.	Highlights advanced feature use that has not been fully explored in Amharic NLP.
This Study	2026	CRF, IB1, IGTree, and SVM for Amharic chunking	Developed a 12,000-sentence IOB2 dataset and compared multiple machine learning models.	Fills gaps in dataset size, model comparison, and lack of CRF/MBL-based Amharic chunking research.

2.8 Summary

Natural Language Processing (NLP) enables computers to understand human language. One key task in NLP is text chunking (shallow parsing) which divides sentences into meaningful parts like noun phrases and verb phrases. It is widely used in applications such as machine translation, question answering, search engines, chatbots and information retrieval systems.

Different methods have been used for chunking. Early rule-based approaches were simple but difficult to scale. Later machine learning models such as HMM, SVM, CRF and MBL were introduced to learn patterns from data. Among them CRF is effective for sequence labeling because it considers word relationships while MBL learns from similar past examples. Studies in other languages show that using linguistic features like POS tags and context improves performance.

For Amharic research is still limited. Previous work mainly used HMM and rule-based methods but performance was restricted due to small datasets, complex sentence structures and issues like label bias.

The literature also shows several gaps limited Amharic NLP resources insufficient use of modern methods like CRF and MBL, small and formal-text-only datasets and weak use of linguistic features in this morphologically rich language.

To address these gaps this study recommends using CRF and MBL with richer features such as POS tags, context and word structure. Future work should focus on larger and more diverse datasets and explore deep learning models like BiLSTM and Transformers.

Overall, improving Amharic text chunking is important for supporting applications such as translation systems, search engines, chatbots and other NLP-based technologies.

CHAPTER THREE

3 DESIGNS AND METHODOLOGY

3.1 Introduction

Amharic is the official working language of Ethiopia and ranks among the most widely spoken Semitic languages, second only to Arabic in number of speakers (Melese et al., 2016). Like other languages it has a rich and systematic grammatical structure that enables clear communication. Its grammar includes major word classes such as nouns, verbs, adjectives, pronouns and adverbs, each playing specific roles in sentence formation. Amharic also forms meaningful phrases like noun phrases, verb phrases and prepositional phrases which combine to build complete sentences. Sentence structures may be simple, compound or complex and the language further uses prefixes and suffixes to express tense, mood and agreement making its morphology highly expressive.

3.2 The Research Design Methodology

This study develops an automatic Amharic text chunking system using a Research Design Methodology implemented through an end-to-end pipeline. It addresses the challenges of Amharic's complex grammar by applying a structured process that includes data collection, text preprocessing, word embedding, machine learning-based modeling, and evaluation. Raw text from sources such as broadcast media, governmental publication and newspapers is first cleaned through normalization, tokenization and POS tagging, then transformed into numerical representations using techniques like Word2Vec and FastText to capture both context and sub-word information. These embeddings are then processed by machine learning models such as BiLSTM-CRF and Memory-Based Learning to identify and group words into meaningful syntactic units like noun and verb phrases. The final output produces structured, labeled chunks that not only represent sentence structure but also serve as a foundation for advanced NLP applications in Amharic, including machine translation and information extraction.

3.2.1 Research Design Method end to end Pipeline

This study follows an end-to-end pipeline based on the Research Design Methodology to process Amharic text and extract meaningful syntactic structures such as Noun Phrases (NPs) and Verb Phrases (VPs). The workflow transforms raw text into structured linguistic information through five main stages they are Raw Amharic text, preprocessing, word embedding, machine learning chunk, chunked output & analysis.

A. **Raw Amharic text** the process begins with collecting raw Amharic text from different sources such as newspapers, broadcast media other written documents. This ensures the dataset reflects real-world language use.

B. **Text Preprocessing** before the system can do the language understanding the text is done through a cleaning and preparation stage that includes character normalization, word tokenization and POS tagging for each word before analysis the text is cleaned and prepared through three steps they are ***Normalization*** is standardizing characters and correcting spelling variations, ***Tokenization*** is Splitting sentences into individual words and ***POS Tagging*** is assigning grammatical labels (such as noun, verb or adjective) providing important syntactic information that supports the chunking process and to each word to support syntactic analysis.

C. **Word Embedding Techniques** After preprocessing text is cleaned every word is transformed and converted into numerical vector representations so that the computer can understand them. Methods like Word2Vec (captures word context) and FastText (captures sub-word and morphological features) are used to represent Amharic words more effectively and Word2Vec or Fast Text which assist in capturing the words' semantics.

D. **Machine learning based chunking** the system then uses having the word as embedding and a machine learning model to identify phrase boundaries and assign chunk labels. BiLSTM (Bidirectional Long Short-Term Memory) and CRF (Conditional Random Field) is used for this task. It is trained on labeled data and learns patterns to correctly segment, detect and label the chunk in the sentences. As an alternative Memory-Based Learning (MBL) is also considered, where the system makes decisions by comparing new sentences with previously stored examples.

E. **Chunked Output & Analysis** in the final step the system produces structured output by dividing sentences into meaningful units such as noun phrases and verb phrases. This helps in understanding sentence structure and supports further natural language processing tasks like translation, information extraction and semantic analysis.

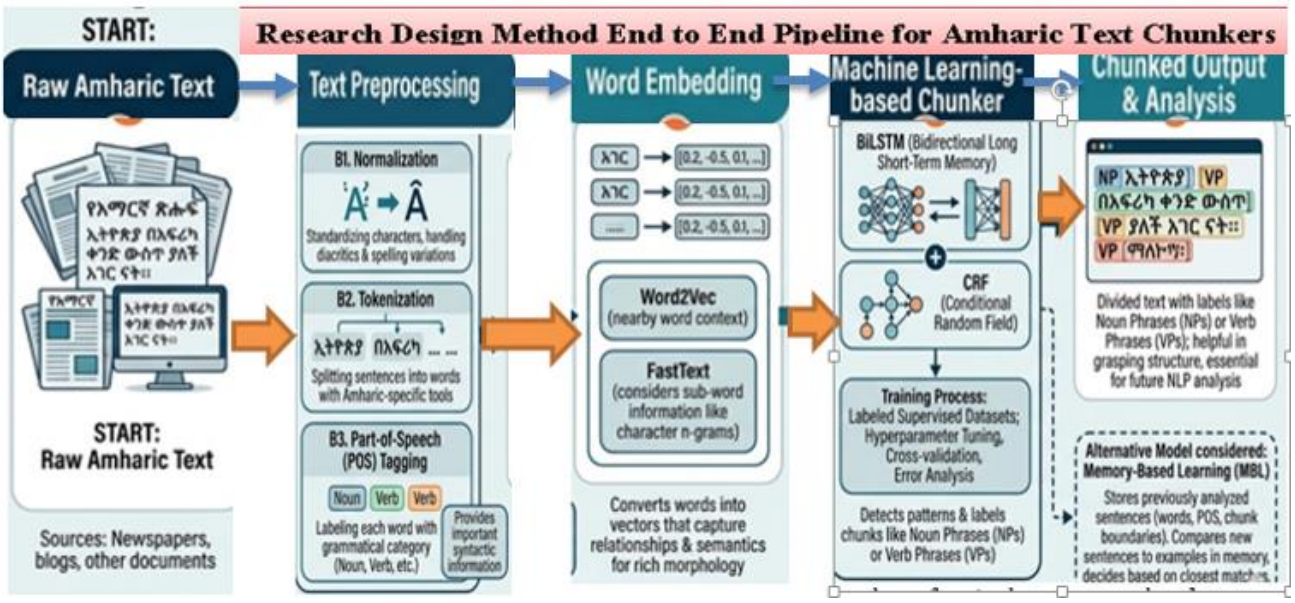


Figure 3.1 Research design method end to end pipeline for Amharic text chunk

3.2.2 Machine Learning Models for Chunking

After embedding the words machine learning models are used to detect patterns and group related words into meaningful chunks like noun or verb phrases the models being explored include

Conditional Random Fields (CRF) a probabilistic model that is especially effective for sequence labeling tasks such as identifying phrase boundaries in text the chunking models will be trained using labeled supervised datasets to ensure optimal performance techniques like hyperparameter tuning**,** cross-validation and error analysis will be employed throughout the development process

Memory-Based Learning (MBL) is a machine learning approach that works much like how people learn from experience. Instead of creating complex rules in advance, the system remembers many examples it has already seen and uses them to make decisions about new data. MBL stores previously analyzed Amharic sentences such as words, their parts of speech, and their correct chunk boundaries. Later, when the system encounters a new Amharic sentence, it compares it with similar examples stored in memory and decides how to chunk the text based on the closest matches

3.3 Data Collection

The Amharic text chunking dataset is collected from multiple sources to capture how the language is used in both formal and informal contexts. These include grammar books and language guides for well-structured Amharic, newspapers and magazines for real-world news and opinions, online content such as blogs, social media and websites for informal writing, broadcast media including Amharic radio and BBC Amharic for spoken-style news content, broadcast media and official government publication documents for formal language usage. Since most of the data is not pre-annotated, it is manually labeled using the IOB2 format to mark phrase boundaries. The annotation process is carefully reviewed with the help of Amharic language experts to ensure accuracy and consistency. After annotation the dataset is split into training and testing sets for model development and evaluation. This curated dataset forms the main foundation for building and testing the Amharic text chunking system.

3.4 Implementation Tools

This study develops an Amharic text chunking system using two supervised machine learning methods: Conditional Random Fields (CRFs) and Memory-Based Learning (MBL). To implement them, two open-source tools are used: CRFsuite for CRF modeling and TiMBL (Tilburg Memory-Based Learner) for MBL. CRFsuite is a widely used toolkit for building CRF models and it is applied here to perform sequence labeling for chunking. TiMBL is a memory-based learning framework and its component Memory-Based Tagger (MBT**)** is used in this study for chunking since it is well suited for tasks like POS tagging and sequence labeling.

CRF models learn patterns by considering the context of surrounding words, while MBL makes predictions by comparing new instances with similar past examples. Combining both methods helps balance contextual learning and example-based reasoning, making them effective for handling the rich morphology and complex sentence structure of the Amharic language.

3.4.1 Model Development Tools Used in the Study

Different tools and techniques are used throughout the development of the Amharic text chunking system. These tools supported dataset collection, corpus preparation, annotation, feature extraction, model training and testing processes.

The annotated datasets are prepared and used to train the CRF and MBL machine learning models. After training the performance of the models was evaluated using common evaluation metrics such as precision, recall, F1-score and accuracy. The evaluation results showed that the type and source of the dataset such as social media, bible and fiction texts have a direct impact on model performance. This indicates that dataset quality and domain diversity are important factors in improving the effectiveness of Amharic text chunking systems.

Table 3.1 Summary of tools and technologies used in this study

No	Category	Used Tools
1	Hardware Tools	-Lenevo Desktop Computer, Processor Intel(R) Core (TM) i5-7200U CPU @ 3.70GHz, 8GB RAM & Windows 10/11 Operating System. -Toshiba Laptop, LAN for internet service
2	Programing Language	-Python, Horn Morpho for morphological analyzer - use the language for ease of integration of the analyzer with our system.
3	Framework/Libraries	Django, Tensorflow, scikit-learn, pandas, NumPy,
4	IDEs and editors	Anaconda3, Jupiter, Notebook, visual studio code
5	Application software	Microsoft Office 2016 Word and Excell, Geez,
6	Browser	-Google chrome, Microsoft edge, Mozilla Firefox,

3.5 Testing and Experimental Procedure

In this thesis conducted a total of twelve experiments to develop and evaluate the Amharic text chunker the first four experiments were carried out using Conditional Random Fields (CRFs) while the remaining eight experiments used Memory-Based Learning (MBL) and carefully prepared the dataset to test our chunker prototype and measured the system's accuracy to assess its performance. After evaluation, the model with the highest accuracy and the best ability to chunk text correctly was selected as the final Amharic text chunker the accuracy of the model was calculated using the formula.

$$\text{Accuracy} = \frac{\text{Number of correctly predicted chunk tags}}{\text{Total number of chunks}}$$

This metric shows how well the system predicts the correct chunk boundaries and labels along with accuracy other evaluation metrics such as precision and recall are often used to get a complete picture of the model's performance ensuring high accuracy is particularly important for languages like Amharic where morphological complexity can make chunking challenging the experiments also helped identify which features and algorithms work best for handling the unique characteristics of Amharic text. F1 score and recall and precision

3.6 Basic Amharic Word Classes

Amharic words are grouped into different grammatical categories known as word classes. Different researchers have proposed different classifications over time. According to (Shiferaw et al., 2024; Upgris & 2021, n.d.; Yimam et al., 2021), Amharic has six major word classes: nouns, verbs, adjectives, adverbs, prepositions, and conjunctions. Later, (Mulu & Goyal, 2020)suggested a simplified classification by combining pronouns with nouns because pronouns often perform the same grammatical function as nouns in sentences. In this study, a hybrid classification approach is adopted by grouping nouns and pronouns together while keeping the other word classes separate, following the structure of the WIC corpus dataset. The discussion is mainly based on studies such as (Upgris & 2021, n.d.)

3.6.1 Noun Class

Nouns are words used to name people, places, objects, animals or ideas. In Amharic nouns can appear in both singular and plural forms. The plural form is commonly created by adding the suffix *"-ኦች"* (-och) to the singular noun. Nouns may also be primitive or derived. Primitive nouns are basic words, while derived nouns are formed from other word classes such as verbs or adjectives (Upgris & 2021, n.d.)

3.6.2 Verb Class

Verbs express actions, events, or states of being. In Amharic sentence structure, the verb usually appears at the end of the sentence, which is one of the important features of the language. Amharic verbs also show information about the subject through suffixes. For example, *"ተኛሁ"* means "I slept," while *"ተኛህ"* means "you slept." Verbs can also be derived by adding prefixes, suffixes, or combining roots with other lexical elements.(Mulu & Goyal, 2020)

3.6.3 Adjective Class

Adjectives are words that describe nouns or pronouns. They provide more information about qualities, appearance, size or condition. In most cases adjectives come before the noun they modify. Example in *"ቆንጆ ሰው"* ("beautiful person"), the adjective *"ቆንጆ"* describes noun *"ሰው."*

3.6.4 Adverb Class

Adverbs modify verbs, adjectives, or other adverbs. They give information about time, place, manner, reason, or degree. For example, in *"እሱ በፍጥነት ሄደ"* ("He went quickly"), the word *"በፍጥነት"* explains how the action was performed.

3.6.5 Preposition Class

Prepositions are words that show relationships such as place, direction, time, or reason. They usually appear before nouns or pronouns and help explain how one part of a sentence relates to another Common examples in Amharic include *"ከ"* ("from"), *"ወደ"* ("to"), and *"ስለ"* ("about" or "for").(Mulu & Goyal, 2020)

3.6.6 Conjunction Class

Conjunctions are words used to connect words, phrases, clauses, or sentences. Common examples include *"እና"* ("and"), *"ወይም"* ("or"), and *"ነገር ግን"* ("but"). Some conjunctions connect equal ideas, while others connect dependent clauses with independent clauses.

3.7 Amharic Phrasal Categories

A phrase is a group of related words that work together to express meaning but do not form a complete sentence. Every phrase is built around a central word called the head word. Additional words may act as modifiers, specifiers, or complements to provide more detail and clarify meaning. In Amharic, phrase categories are generally based on the major word classes.

3.7.1 Noun Phrase (NP)

A noun phrase is built around a noun or pronoun and may include modifiers or specifiers that provide additional information. Noun phrases can be simple or complex Examples include *"ቤት"* ("house") and *"ትኩስ እንጀራ"* ("hot injera").

3.7.2 Pronoun Phrase

A pronoun phrase is centered on a pronoun such as "I," "you," or "he." These phrases commonly function as subjects or objects in sentences and are important for identifying grammatical roles in chunking systems

3.7.3 Verb Phrase (VP)

A verb phrase is built around a verb and may include objects, complements, or modifiers. Some verb phrases require objects to complete their meaning, while others do not Verb phrases may also be simple or complex depending on their structure.

3.7.4 Adverbial Phrase (AdvP)

An adverbial phrase is centered on an adverb and gives additional information about time, place, manner, degree. Example *"በፍጥነት"* ("quickly") describes how an action happens.

3.7.5 Adjectival Phrase (AdjP)

An adjectival phrase is built around an adjective and may include modifiers or specifiers that strengthen or clarify meaning For example, *"በጣም ብልህ"* ("very smart") is an adjectival phrase where *"በጣም"* intensifies the adjective.

3.7.6 Prepositional Phrase (PP)

A prepositional phrase begins with a preposition and is usually followed by a noun or noun phrase. These phrases commonly express location, direction, or time. For example, *"በቤቱ ውስጥ"* means "inside the house."

3.7.7 Interjection Phrase

Interjection phrases express emotions such as surprise, happiness, pain, or excitement. Words like *"ዋይ!"* and *"እውነት!"* are common examples. These phrases are useful in NLP applications because they help identify emotions and speaker attitudes in text.

3.7.8 Determiner Phrase (DP)

Determiner phrases contain words that identify or limit nouns, such as demonstratives, possessives or numerals .Examples include *"ይህቤት"* ("this house") and *"የእኔ ጓደኛ"* ("my friend").

3.7.9 Particle Phrase

Particle phrases include small grammatical words used for emphasis, negation, or sentence focus. Examples include *"ነው"* and *"እንጂ"*. These phrases are important in Amharic text chunking because they help identify sentence meaning and phrase boundaries accurately.

3.8 Amharic Sentence Types

In Amharic grammar a sentence is formed by combining words and phrases to express a complete idea. A sentence may express a statement, question, command or emotion. Most Amharic sentences contain a subject, usually a noun phrase (NP) and a predicate, usually a verb phrase (VP).

For example, in the sentence *"አበበ በትክክል ሥራውን አጠናቀቀ"* ("Abebe completed his work properly"), *"አበበ"* is the subject, *"ሥራውን"* is the object, *"በትክክል"* modifies the action, and *"አጠናቀቀ"* is the verb. Based on their structure, Amharic sentences are commonly classified as simple, compound, complex, and compound-complex sentences.

3.8.1 Simple Sentences

A simple sentence contains only one main verb and expresses a single idea .It is usually formed from a noun phrase followed by a verb phrase. For example, *"አስቴር ብርጭቆውን ሰበረችው"* ("Aster broke the glass") contains only one verb, *"ሰበረችው."* Simple sentences may include intransitive verbs, transitive verbs with one object, or verbs with two objects.

3.8.2 Compound Sentences

A compound sentence contains two or more independent clauses connected by coordinating conjunctions such as *"እና"* ("and"), *"ነገር ግን"* ("but"), or *"ወይም"* ("or") .

For example, *"እሱ ቤት ሄደ እና እሷ ገባች"* ("He went home and she entered"). Each clause can stand alone as a complete sentence. Compound sentence recognition is important in Amharic text chunking and parsing systems.

3.8.3 Complex Sentences

Complex sentences contain dependent clauses, embedded clauses, or complex noun and verb phrases these structures make sentences richer and more detailed and are important in NLP tasks such as phrase chunking and shallow parsing .Example, *"ልጁ ማንነቱን ለማረጋገጥ ሞከረ"* ("The boy tried to verify his identity") contains a complex verb phrase. Amharic also commonly uses embedded clauses and multi-verb constructions, which help express detailed meanings within a sentence. Such sentence patterns are important in machine learning models such as CRFs and HMMs used for Amharic text chunking.

3.8.4 Compound-Complex Sentences

A compound-complex sentence combines at least two independent clauses with one dependent clause. These sentences are formed using both coordinating and subordinating conjunctions. Example *"ምክንያቱም ሞቃት ነበር፣ እሱ ቤት ሄደ እና እሷ በውሃ ተቀባች"* ("Because it was hot, he went home and she took a shower"). This type of sentence expresses multiple related actions and cause-and-effect relationships. It is especially useful in text chunking and sequence labeling systems such as CRFs and HMMs.

3.8.5 Declarative Sentences

Declarative sentences are used to state facts, ideas, or information clearly. They are the most common type of sentence in both Amharic and English. In Amharic, they usually end with the punctuation mark

"።" For example, *"አበበ የመጀመሪያ ክፍል ተማሪ ነው።"* ("Abebe is a first-grade student.").

3.8.6 Negative Sentences

Negative sentences express that something did not happen or is not true. In Amharic, negation is commonly formed using the prefix *"አል-"* and the suffix *"-ም"* For example, *"ሚካኤል እንጀራ አልበላም"* means "Mikael did not eat injera."

3.8.7 Interrogative Sentences

Interrogative sentences are used to ask questions and usually end with a question mark. They may ask about a person, place, object or action Example, *"ምን ታደርጋለህ?"* ("What will you do?"). Common interrogative words in Amharic include *"ማን"* ("who"), *"ምን"* ("what"), and *"የት"* ("where").

3.8.8 Imperative Sentences

Imperative sentences are used to give commands, instructions, or requests. In many cases, the subject is omitted because it is already understood from the verb form. For example, *"እሱ ይሂድ!"* means "Let him go!"

3.8.9 Exclamatory Sentences

Exclamatory sentences express strong emotions such as surprise, excitement, anger, admiration. They often include emotional expressions or interjections like *"እውነት!"* or *"ዋይ!"* For example, *"እውነት እሱ አሸንፏል!"* means "Indeed he has won!" These expressions are important in Amharic text chunking because they may indicate emotional or emphasis-based phrase boundaries.

3.9 Proposed Models of the Study

This study proposes an automatic Amharic text chunking system developed using supervised machine learning approaches mainly Conditional Random Fields (CRF) and Memory-Based Learning (MBL)

The system is designed to automatically identify meaningful phrase structures in Amharic sentences such as noun phrases, verb phrases, adjective phrases and prepositional phrases. Because Amharic is a morphologically rich language with complex grammatical structures identifying phrase boundaries manually is difficult and time consuming.

The proposed model follows an end-to-end processing architecture that transforms raw Amharic text into structured chunked output through several connected stages see below diagram.

The workflow begins by collecting raw Amharic text from newspapers, grammar books, broadcast media and government publications using multiple sources helps the system learn both formal and informal language patterns. After data collection the text passes through a preprocessing stage. During preprocessing stage the text is cleaned through normalization, tokenization and Part-of-Speech (POS) tagging. Normalization standardizes different Amharic character variations tokenization divides

sentence into individual words and POS tagging assigns grammatical labels such as noun, verb, adjective and pronoun to each word. These preprocessing steps prepare the text for machine learning analysis and improve chunk prediction performance

Next, the system performs feature extraction and word representation. Important linguistic and contextual features such as word tokens, neighboring words, POS tags and contextual information are extracted from the text. Word embedding techniques like Word2Vec and FastText are also used to convert words into numerical vector representations. Word2Vec captures contextual meaning between words, while FastText captures sub-word and morphological information which is important for handling the complex structure of the Amharic language.

The processed data is then passed to the machine learning-based chunking stage. The primary model used in this study is Conditional Random Fields (CRF) a probabilistic sequence labeling algorithm that predicts chunk labels by considering relationships between neighboring words and labels in a sentence. Unlike models that process words independently CRF analyzes the entire sentence structure making it highly effective for text chunking tasks.

Alongside CRF Memory-Based Learning (MBL) is also used as an alternative chunking approach. MBL works by storing previously learned examples and comparing new input sentences with similar examples seen during training. In this study IB1 and IGTree algorithms are used as the main MBL techniques for chunk classification. This approach allows the system to learn directly from examples instead of depending only on generalized statistical rules.

The chunk prediction process begins after preprocessing and feature extraction are completed. Each word is represented using multiple linguistic and contextual features. The CRF model predicts whether a word belongs to the beginning, inside or outside of a chunk using the IOB2 chunk tagging scheme. Similarly, the MBL model predicts chunk labels by comparing new sentences with previously stored examples and selecting the closest matching patterns.

The implementation of the proposed system uses CRFsuite for CRF modeling and TiMBL for Memory-Based Learning implementation. These tools support model training, feature extraction, sequence labeling and chunk prediction.

The input to the proposed system consists of raw Amharic sentences collected from different textual sources. Before chunk prediction the text undergoes preprocessing, feature extraction and embedding representation. The final output of the system is a structured chunked sentence labeled using the IOB2 format. Each word receives chunk labels such as B-NP, I-NP, B-VP or O, which identify phrase boundaries and syntactic structures within the sentence. The generated output can support advanced Natural Language Processing applications such as machine translation, information extraction,

semantic analysis, parsing and intelligent Amharic language processing systems. See workflow diagram and Mathematical model below in detail brief.

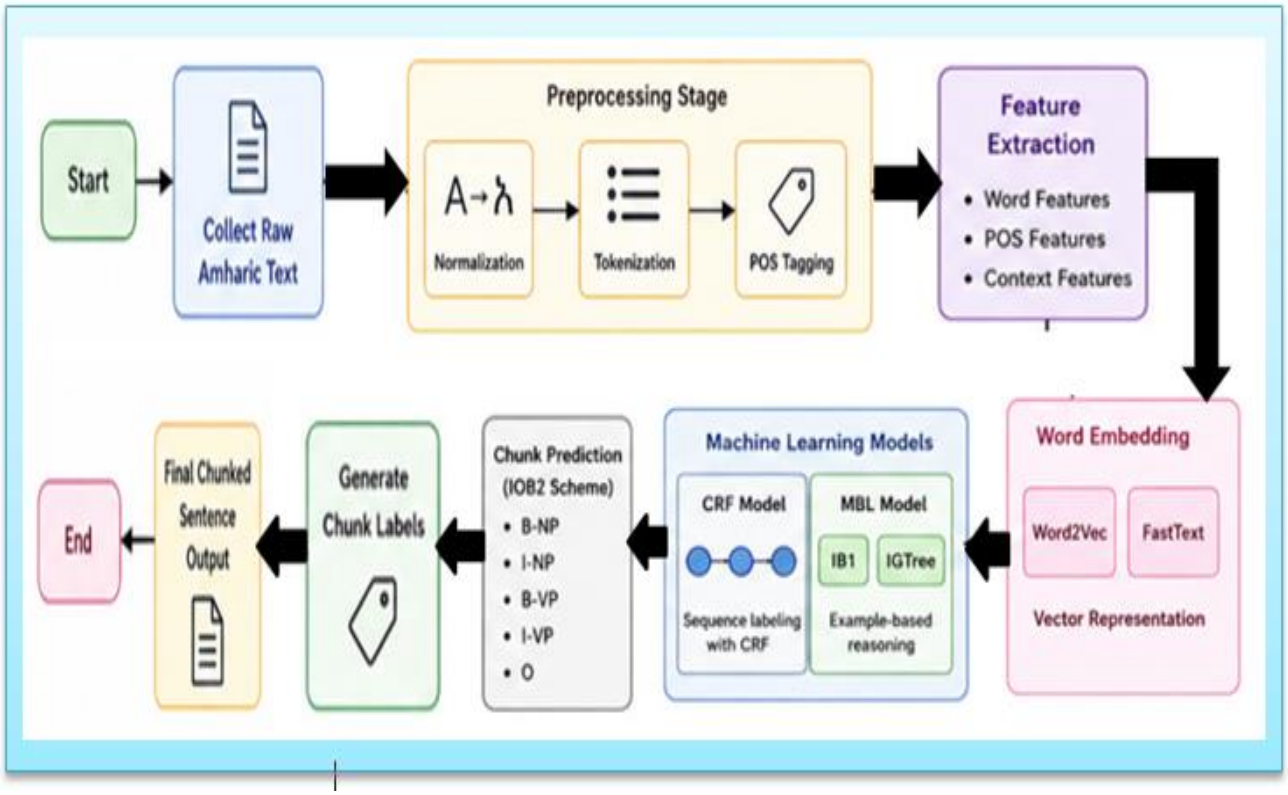


Figure 3. 2 Workflow diagram of proposed model

CHAPTER FOUR

4. AMHARIC TEXT CHUNKER MODEL

4.1. Introduction

This chapter is to provide a detailed description of the overall system design and process of preparing the dataset that is used for training and testing for Amharic text chunk. A text chunk is a system that identifies and groups words in a sentence into meaningful units such as noun phrases or verb phrases. Developing such a system requires proper preparation of data and a model architecture that reflects the linguistic characteristics of a language this chapter is categorized into two main parts they are dataset preparation and system design model.

4.1.1 Dataset Preparation

Dataset preparation describes how data was collected, organized and processed to serve as input for the model. Preparing the dataset involves several procedures that include cleaning text, labeling each word with the correct chunk information and splitting data into training and testing sets. Proper dataset preparation ensures the model can learn patterns effectively and be accurately evaluated.

As discussed in the previous chapter, the design of an Amharic text chunk using machine learning needs a syntactically annotated corpus in natural language processing (NLP). The success of any application like a text chunk depends heavily on having a high-quality dataset and carefully selecting the right linguistic features in computational linguistics a dataset is often called a corpus (plural corpora) The Corpus can be divided into two sections annotated corpus and unannotated corpus.

- **Annotated Corpus** is texts with linguistic labels such as Part-of-Speech (POS) tags, chunk tags or named entity tags (NER) and **Unannotated Corpus** is raw texts without any syntactic or linguistic labels.

This study follows a supervised machine learning approach. An annotated corpus is required for every machine-learning-based text chunk. It needs two types of datasets, such as training and testing dataset.

- **Training dataset** is to teach/learn the model to recognize and determine the patterns in the language. And **testing dataset** evaluates the performance of the trained model on unseen data.

A dataset was prepared specifically for this study using a total of 12000 Amharic sentences collected from government documents and broadcast media. Some sentences from broadcast media already had POS tags but most sentences from books and magazines have not been tagged. These were manually POS-tagged by the researchers and reviewed by a linguistic expert for accuracy. However, POS tagging alone is not enough to train a chunkier. The dataset also requires chunk tags, which mark the boundaries of phrases such as noun phrases (NP) or verb phrases (VP), since no chunk-tagged corpus exists for Amharic; all sentences were manually chunk-tagged by the researchers in this study.

- Used dataset data definition and its descriptive statistics the study.

First 5 rows of the dataset:

	Sentence ID	Token	POS	Chunk
0	1	በሚዛን	Noun Phrase	B-NP
1	1	አማን	Noun	I-NP
2	1	ክተማ	Noun	I-NP
3	1	አስተዳደር	Noun	I-NP
4	1	በ2017	NUMCR	B-NP

```
Information about the DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12000 entries, 0 to 11999
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Sentence ID  12000 non-null  int64
 1   Token        12000 non-null  object
 2   POS          12000 non-null  object
 3   Chunk        12000 non-null  object
dtypes: int64(1), object(3)
memory usage: 375.1+ KB
```

Descriptive statistics:

	Sentence ID	Token	POS	Chunk
count	12000.000000	12000	12000	12000
unique	NaN	4659	37	13
top	NaN	ላይ	Noun	I-NP
freq	NaN	106	6487	4827
mean	3.911750	NaN	NaN	NaN
std	2.658894	NaN	NaN	NaN
min	1.000000	NaN	NaN	NaN
25%	2.000000	NaN	NaN	NaN
50%	3.000000	NaN	NaN	NaN
75%	5.000000	NaN	NaN	NaN
max	22.000000	NaN	NaN	NaN

Checking for missing values:

	0
Sentence ID	0
Token	0
POS	0
Chunk	0

dtype: int64

Figure 4. 1 Dataset used by the study and its description.

- POS tagging used in this study is number of 37 tags (See description on Appendix 2).
- Chunk tagging used by this study is 13 tags (See the description on Appendix 3).
- Encoding all sentences has been transcribed using the Amharic alphabet Unicode standard (see Appendix 1) and the corpus is carefully reviewed and feedback from a linguistic expert was incorporated to ensure quality and correctness.

4.1.2. Overview of the Proposed System Design Model

This section explains the structure of Amharic text chunk developed in this study. It describes the system components, how they interact and the overall flow of information from input text to chunked output. This includes machine learning techniques used for feature representation and how the model decides which words belong to which chunks as previous chapters discussed the fundamental aspects of the Amharic language, such as word classes (nouns, verbs), phrase categories (noun phrases, verb phrases) and sentence structures. These linguistic features are essential for designing a chunking system because they provide the patterns and model must learn related work on text chunking for other languages that also informed the design of this model.

This study uses a supervised machine learning approach, meaning the model learns to predict chunks from labeled examples in the dataset and the following sections provide a detailed description of the feature selection training process and evaluation methods used to measure the model's performance. Generally, this section is a bridge to theoretical understanding of the Amharic language with the practical procedure of building a text chunk showing how linguistic knowledge and machine learning techniques work together to create an effective system.

4.2. Chunk Boundary Identification Using IOB2 Tagging Scheme

To identify chunks in a sentence, the system requires you to know where one phrase ends and another begins this is done by using Part-of-Speech (POS) tags assigned to each word. POS tags provide useful grammatical information that helps the system understand how words naturally group together to form phrases such as noun phrases and verb phrase.

There are several standard ways to represent chunk boundaries in text, namely IOB1, IOB2, IOE1, and IOE2. Each method marks chunk boundaries in a slightly different manner. In this study, the IOB2 tagging scheme was selected because it is simple, clear and widely used in text chunk tasks. In the IOB2 scheme, each word is assigned one of the following tags:

- B (Begin) marks the first word of a chunk.
- I (Inside) am marked words that continue the same chunk.
- O (Outside) is marked words that do not belong to any chunk.

The essential benefits and advantages of IOB2 are: -

- Tagging scheme has made it easier for both machines and people to understand language.
- It is easy to spot chunk boundaries, which mean initially, every chunk starts with a "B" tag. It is simple to indicate where one chunk ends and another begins.
- It handles words next to each other properly. If you have two chunks of the same type of side by side, the "B" tag clearly indicates the start of each one, so there is no confusion.
- It helps machines learn better because of clear labeling and makes it easier for machine learning models to recognize patterns and understand where chunks are during training.
- It is user-friendly for humans. When people look at the tagged data, it is straightforward to see the structure of sentences right away.
- It is flexible with different chunk types and, whether it is noun phrase or verb phrase, the system keeps things organized without mixing them up.
- The following Amharic sentence is indicating how the IOB2 works to consider and recognize its advantages and the above benefits **በሚዛን-አማን ከተማ አስተዳደር 20 ሺህ ማህበረሰብ ለብሔራዊ ምክክር ተዘጋጀ** "In Mizan-Aman City Administration twenty thousand communities are ready for national dialogue." Using the IOB2 scheme the sentence is divided into meaningful chunks as shown in the following table.

Table 4.1 Chunk Specification Using IOB2

Amharic Word	IOB2 Chunk
በሚዛን-አማን	B-NP
ከተማ	I-NP
አስተዳደር	I-NP
20	B-NP
ሺህ	I-NP
ማህበረሰብ	I-NP
ለብሔራዊ	B-PP
ምክክር	B-NP
ተዘጋጀ	B-VP

English Sentence	IOB2 Chunk
In	B-PP
Mizan-Aman	B-NP
City Administration	I-NP
Twenty	B-NP
Thousand	I-NP
Communities	I-PP
Are	B-VP
Ready	I-VP
for	B-PP
National	B-NP
Dialogue	I-NP

4.2.1. Description of chunk specification using IOB2

- The first section of the sentence the words "**በሚዛን-አማን ከተማ አስተዳደር**" are grouped together as noun phrase, so the first word is marked with B-NP and the following words with I-NP this shows that they belong to the same phrase.
- The second section expression "20 **ሺህ ማህበረሰብ**" forms another noun phrase since it starts with a new phrase. It is again marked with B-NP followed by I-NP for the remaining words.
- The last section is "**ለብሔራዊ ምክክር ተዘጋጀ**" form a verb phrase. The first word of this phrase is labeled B-VP and the remaining words are labeled I-VP, indicating that they continue the same verb phrase. By labeling words in this way, the chunk can easily recognize phrase boundaries and understand the basic structure of the sentence. This clear and consistent representation is especially helpful when training supervised machine-learning models for Amharic text chunking.

4.3 Proposed Amharic Text Chunk Model Architecture

The proposed Amharic text chunking model is developed to automatically identify meaningful phrase structures such as noun phrases and verb phrases from Amharic sentences. The system follows a machine learning-based architecture that processes raw Amharic text step by step until it produces chunked sentence structures. The architecture consists of five main layers Input Layer, Preprocessing Layer, Feature Extraction and Word Embedding Layer, Machine Learning Chunking and Output Layer

4.3.1 Input Layer

The input layer is the starting point of the system in this stage raw Amharic text is collected from different sources such as newspapers, government publications, broadcast media and online documents. These sources help the system learn different writing styles and sentence structures used in the Amharic language the collected dataset contains about 12,000 manually annotated Amharic

sentences prepared using the IOB2 chunk tagging format. Each sentence includes words, Part-of-Speech (POS) tags and chunk labels that are used for training and testing the machine learning models.

4.3.2 Preprocessing Layer

The Preprocessing Layer prepares the raw Amharic text for machine learning analysis. Since Amharic is a morphologically rich language with complex writing patterns, preprocessing plays an important role in improving the quality and consistency of the dataset the preprocessing stage consists of several important tasks:

4.3.2.1 Text Normalization

Normalization standardizes different character representations and removes inconsistencies in the text. In Amharic writing, some characters may appear in multiple forms while representing the same sound or meaning. This stage converts such variations into a consistent representation. It also removes unnecessary symbols, extra spaces and formatting issues that may negatively affect model performance.

4.3.2.2 Tokenization

Tokenization divides sentences into individual word units called tokens. Each word in a sentence becomes a separate token that can later be analyzed independently. Tokenization allows the system to process sentence structures step by step and identify the relationships between neighboring words. For example “አበበ ወደ ቤት ሄደ” is divided into [“አበበ”, “ወደ”, “ቤት”, “ሄደ”]

4.3.2.3 Part-of-Speech (POS) Tagging

In this stage each token is assigned a grammatical category such as noun, verb, adjective, pronoun or preposition. POS tagging provides important syntactic information that helps the chunking model understand sentence structure for example አበበ → Noun (N) and ሄደ → Verb (V) POS information is one of the most important linguistic features used by CRF and MBL models for predicting chunk boundaries accurately.

4.3.2.4 IOB2 Chunk Annotation

The processed tokens are labeled using the IOB2 chunk tagging scheme. The labels indicate whether a word is at the Beginning (B), Inside (I) or Outside (O) of a chunk example አበበ B-NP, ወደ B-PP, ቤት I-PP and ሄደ B-VP this representation enables the machine learning algorithms to learn chunk patterns from annotated examples.

4.3.3 Feature Extraction and Word Embedding Layer

In this layer important linguistic and contextual information is extracted from the text. The system uses features such as the current word, neighboring words, POS tags, prefixes, suffixes and morphological

patterns. These features help the model understand the relationship between words in a sentence the system also applies word embedding techniques such as Word2Vec and FastText to convert words into numerical vector representations. Word2Vec captures contextual meaning between words while FastText captures sub-word and morphological information which is especially useful for the complex structure of the Amharic language these representations help the machine learning models better understand semantic and contextual relationships between words.

4.3.4 Machine Learning Chunking Layer

This is the main processing layer of the proposed system in this stage the extracted features and word embeddings are used by machine learning algorithms to predict chunk labels for each word in the sentence the study mainly uses Conditional Random Fields (CRF) and Memory-Based Learning (MBL) models such as IB1 and IGTree.

The CRF model predicts chunk labels by considering the relationship between neighboring words and labels across the entire sentence this helps the system identify chunk boundaries more accurately.

The MBL model works differently by storing training examples and comparing new words with previously seen examples. It predicts chunk labels based on similarity measures such as Overlap and Modified Value Difference Metric (MVDM) and these machine learning models learn chunking patterns from the annotated dataset and classify each word into its appropriate chunk category.

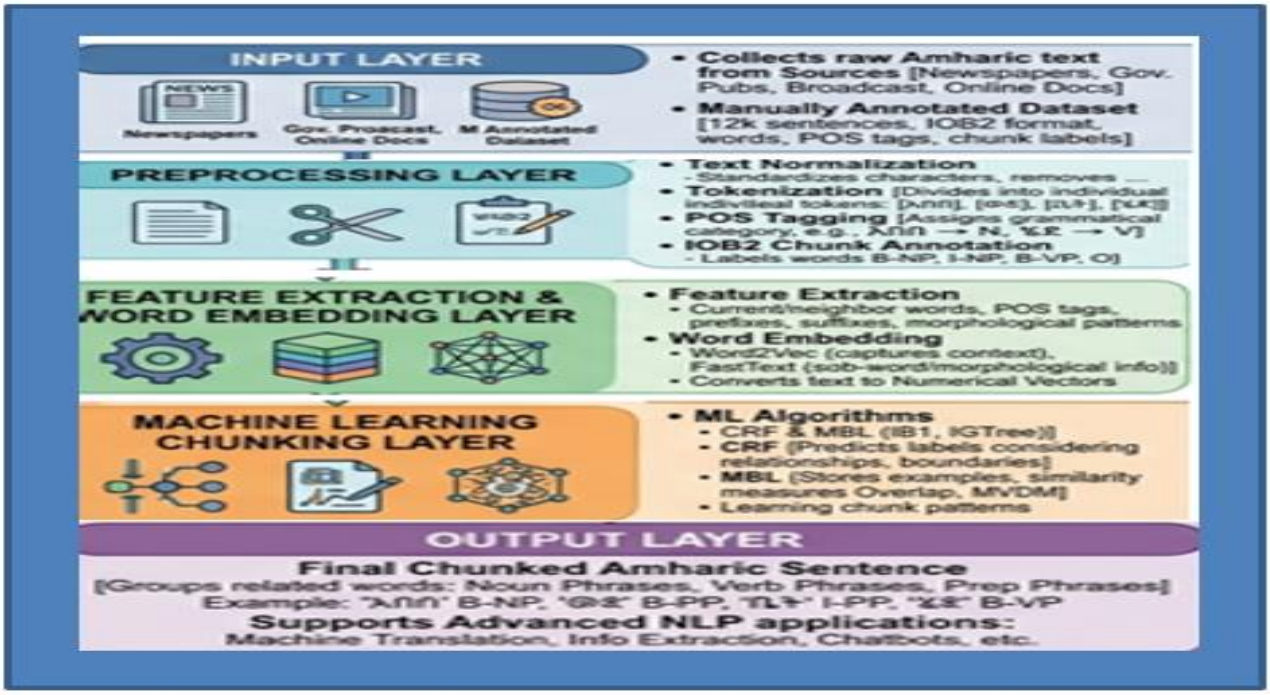


Figure 4.2. Proposed ATC Model Architecture Diagram

4.3.5 Output Layer

The output layer generates the final chunked Amharic sentence. After classification each word receives a chunk label such as B-NP, I-NP, B-VP or O the system then groups related words into meaningful

syntactic units such as noun phrases, verb phrases and prepositional phrases like '*አበበ' B-NP, 'ወደ' B-PP, 'ቤት' I-PP and 'ሄደ' B-VP* the final chunked output can support advanced Natural Language Processing applications such as machine translation, information extraction, speech recognition, chatbots, search engines and question answering systems.

4.4 Architecture of CRFs Based Amharic Text Chunk

The CRFs-based Amharic text chunk is designed to learn from examples first and then make predictions. It works in two main processes learning (training) and prediction (testing) and Amharic Text. Chunk is intended as a means that it primarily learns properties and parameters associated with chunk tags from the training data. It then receives input word with its POS tag and predicts the possible chunk tags. The architecture has two processes the learning process and the prediction process.

- During the learning process the system is trained using an annotated Amharic corpus. First the training data is preprocessed and chunk tags are assigned using the IOB2 format then the Amharic Text Chunk (ATC) encoder reads each word together with its POS tag and chunk tag. After that the feature extractor selects important linguistic features from the words and tags. These features are sent to the model builder which learns patterns from the data and produces a trained CRs model.
- In the Prediction process, the trained model takes new Amharic sentences with POS tags and automatically predicts the correct chunk tags for each word.

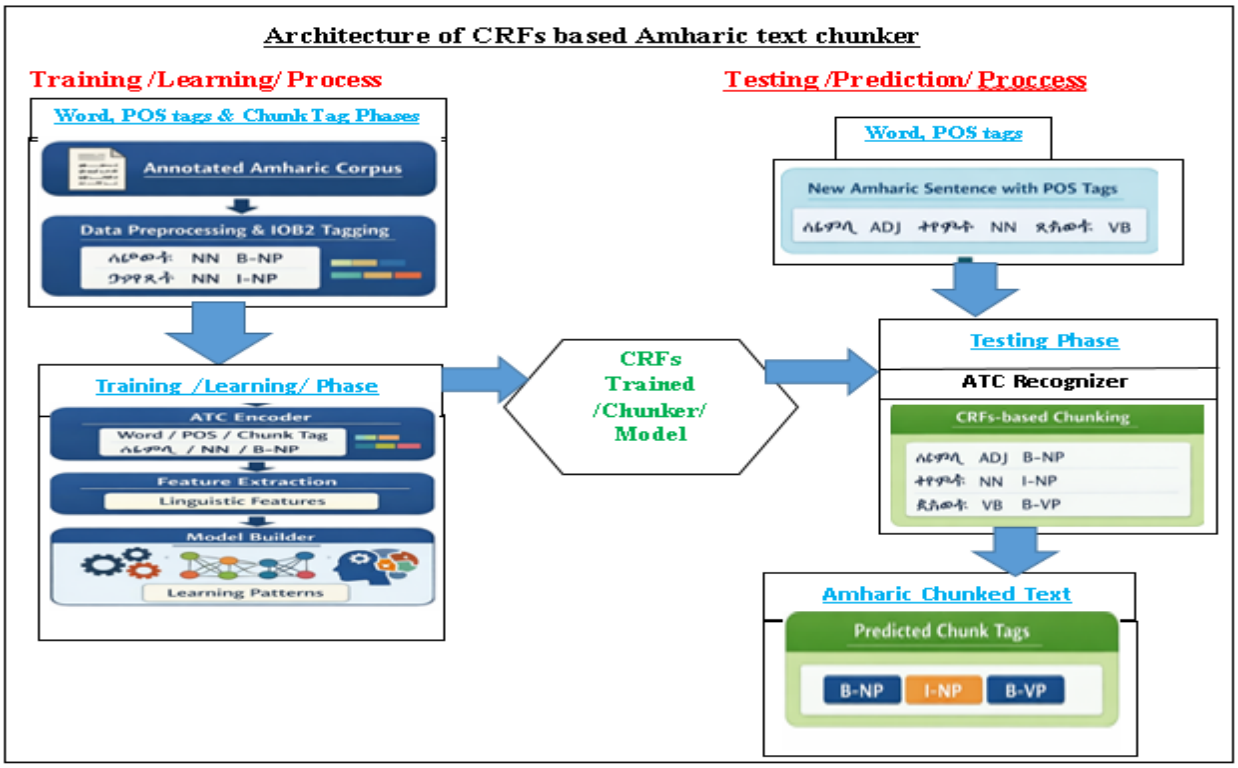


Figure 4. 3 Architecture of CRs based on Amharic text chunk.

This diagram shows how Amharic Text Chunking system learns and uses what it learned to label new Amharic text.

- On the left side it is in teaching the system (learning) stage it is given three things together the Amharic word, its POS (Part of Speech) tag and its correct chunk tag. On the primary side. The ATC Encoder reads and organizes this information then the Feature Extractor looks deeper and pulls out useful patterns and linguistic features. Then after that the Model Builder uses these features to train and form a strong CRF chunk model. By the end of this phase the system has learned how Amharic words should be chunked based on examples.
- The right side is the testing (prediction) stage. Now the system is given new Amharic text but this time it only receives the word (new Amharic sentences with POS tag) and its POS tag. These go into the ATC Recognizer (CRFs-based chunking) which uses the previously trained CRF model to figure out the correct chunk tags. Finally the system produces Amharic chunked text (predicted chunk tags) as output, meaning each word is now labeled with the proper chunk.

4.4.1 CRF Training Process

CRFs Training Method the Training Phase covers every required component employed in the Amharic text chunk's machine learning component training process. At this stage the system runs right on data gathered from training corpus. The primary component is Amharic text chunk encoder, feature extractor, model builder and each of them is essential for producing a dependable trained model.

First, the Amharic text chunkier encoder processes the training data and is essential for accurately arranging the material. It painstakingly finds every token from the corpus together with its related tag. The encoder determines which section of the data is a token and which is a tag then sends this right and well-structured data to the following process. This obligation makes the encoder especially vital. If it wrongly identifies tokens or tags the next steps will also get wrong information and this could finally result in construction of a weak or wrong model. Therefore it is very important to keep the corpus and the Amharic text chunkier encoder in perfect sync to guarantee that the data forwarded is always correct.

Secondly the Feature Extractor starts as was already covered. Properties of text that offer helpful information related to chunk tags are the characteristics. These elements enable the system to raise its confidence in forecasting every word's proper chunk tag. The Amharic text chunkier Feature Extractor painstakingly finds and pulls all required elements from the training data. It is one of the most important parts as it gives the system the crucial information required for model building. The feature extractor is meant not just to pull these characteristics but also to correctly preserve them so that they might be applied optimally during model development and later during prediction.

The Model Builder finally executes its primary responsibilities. The machine learning component seeks most essentially to create a properly trained model fit for usage during the forecasting stage. To produce this trained model the Model Builder uses the data readied by the encoder and the features provided by the feature extractor. A robust and well-trained model will enable improved prediction and more precise chunk tag results as the system handles fresh data in future.

4.4.2. CRF Testing Process

The testing phase is the final phase of the Amharic text chunk system. It is the process of recognizing a chunk tag of the given preprocessed tokens. The knowledge in the model builder containing features that are extracted and stored during the training phase is supplied to the recognizer to identify the chunk tag from the given Amharic text. Identification of the chunk tag is performed based on the calculated probability which is discussed in chapter two.

4.5 Architecture of MBL-based Amharic Text chunk

The Memory-Based Learning Text Chunk operates similarly to Conditional Random Fields (CRF) and consists of two key phases: A training phase and a testing phase. This structure allows the system to analyze and classify text effectively making it particularly suitable for languages like Amharic.

4.5.1. MB Training Phase

It is the foundation of an essential baseline for the system that the training phase serves as the backbone of the Amharic text chunk. It establishes the groundwork necessary for the system to understand and process text accurately and think of the training phase as the foundation of a building. It needs to be solid for the building (the chunk) to stand strong for the Amharic text. This phase is crucial to help the system understand language patterns.

- It has feature extraction during this phase the text chunk employs feature extraction to prepare the input words for memory-based learning. This involves analyzing the text data to create manageable instances by utilizing a used technique called windowing. The system examines fixed-length segments of text by looking at a target word along with a set number of words before and after it. This windowing approach helps capture the context surrounding each word.
- Uses of instance creation as the system analyzes the annotated words it automatically extracts instances. Each instance is a snapshot comprising a target word along with its surrounding context. This systematic extraction helps the MBL model learn and build a robust dataset enabling it to recognize patterns in language. Once the annotated words are provided the system automatically extracts instances from the dataset. Each instance comprises a target word and its surrounding context allowing the model to learn language patterns effectively.

- Using model builder and development in these instances the MB model (TiMBL) builds a predictive model. This model will be used during the testing phase to classify new text inputs and, utilizing these instances the MB system (like TiMBL) constructs a model. This model serves as a reference for predicting chunk tags during the testing phase.

4.5.2. MBL Testing Phase

MB Testing Phase It compiles and keeps testing phase information. Like a student remembering previous lessons the system stores training data representations in memory. This saved data helps one to classify fresh sentences during testing. The method keeps important representations from the training stage and images it as a student who has learned some passages from their books. These memorized items will help them as they approach fresh inquiries or sentences.

New cases are categorized and the system is extrapolated to handle fresh or strange words by means of memory stored example comparison. Extrapolation is the process by which the model searches for the most comparable stored cases to help it to know how to tag the fresh data; during testing the model searches for similarities much like one would for a well-known story among new ones. The similarity of measurements that are distinct methods inside the MB framework varies mostly depending on their definition of similarity. This suggests they could remember and recall events from memory in different ways. Algorithms included into TiMBL differ primarily in their definitions of similarity which affects how instances are stored and the memory is searched in categorization this variance lets the system change according to context. It now analyses fresh cases. The program looks at every fresh sentence broken down into instances should the characteristics of a new case match an already present one in memory the method can reliably allocate the appropriate chunk tag and when fresh sentences arrive the technology breaks them into controllable chunks. It then searches memory for the nearest equivalents. Should it find a perfect match it can certainly project the chunk tag for the fresh example. The model uses an analogy method if there is no exact match. Based on shared characteristics it derives from this related case it locates the memory instance most like it then uses it to categorize the fresh input. The classifier continuously checks the fresh instance against every entry kept in memory in the training set. This thorough search guarantees that the system locates the best match therefore enabling it to make wise judgments on chunk tagging.

Robustly combining training and testing phases the MB (Memory-Based Learning Text Chunker) helps it to properly grasp and classify material using a memory-based method it handles the complexity of language especially in languages like Amharic therefore turning it into a great tool for text analysis. The chunker effectively negotiates the subtleties of language via its methodical feature extraction and comparison approaches so enabling it to provide correct classifications and insights in the MB text chunk, it combines memory storage with a sophisticated technique of comparison to categorize and learn from language data. It successfully divides and tags text into significant units by means of parallels between its training and testing procedures hence improving its grasp and handling of the Amharic language this combination of learning from experience and application of that knowledge is what makes memory-based learning powerful and adaptable

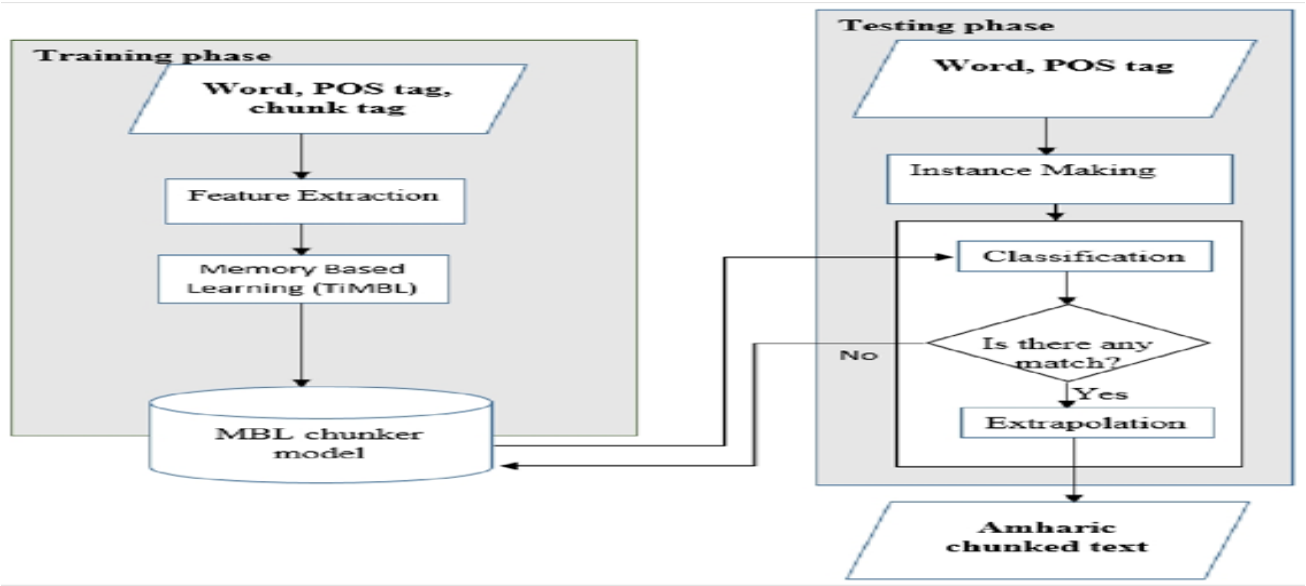


Figure 4.4 Architecture of Memory-Based Learning of ATC.

4.6 Applications of the ATC

By dissecting Amharic sentences into short significant word groupings Amharic text chunking a technology helps computers grasp them. The computer learns which words go together such names, actions, places or descriptions instead of analyzing each word independently. To teach the machine how to automatically identify these groups from examples machine learning algorithms like Conditional Random Fields (CRF) and Memory-Based Learning (MBL) are applied. This helps the computer to be far more intelligent when handling the Amharic language machine translation the system first has to grasp the sentence's structure while converting Amharic into English or another language. Chunking enables the machine to determine which words comprise a whole idea together. For instance the program interprets "አበበ" as the person "ወደ ቤት" as the location phrase and "ሄደ" as

the action in the phrase “አበበ ወደ ቤት ሄደ” This knowledge helps the translation to be more grammatically sound and natural.

These are meant to gather crucial information from documents websites or news stories. The computer uses chunking to quickly identify significant phrases like people's names, places, organizations and events instead of reading material like a human would. In an Amharic news story for instance the technology can instantly identify terms about government buildings, cities or public figures therefore guiding consumers in search and analysis of data more readily.

Speech recognition programmers Speech recognition software transform spoken word into text when someone talk in Amharic if the machine only views independent words though interpreting the sentence properly is challenging by organizing the words into significant unit chunking enable the system to grasp the speaker meaning more fully this raises the correctness and quality of voice-based apps

Programs that respond to inquiries these systems help users to find solutions to their questions. By chunking the computer may find the right answer from stored data and grasp the primary subject of the question. For instance should someone inquiry "የኢትዮጵያ ዋና ከተማ ምንድነው?" the system finds the crucial phrase "የኢትዮጵያ ዋና ከተማ" and properly replies "አዲስ አበባ."

Chunking helps grammar checking systems as well writing Amharic sentences many authors and students make grammatical errors. Chunking helps the machine grasp sentence structure and find mistakes more precisely. Then educational tools can advise corrections and enable users to develop their writing ability.

Long text summarization is a good tool for readers to grasp too much information fast Amharic documents or news reports might include too much. Chunking enables the computer to automatically produce brief and significant summaries by highlighting the most significant phrases and concepts.

Digital libraries and search engines also utilize chunking to help to improve search results the system grasps whole phrases instead of searching particular words individually. When searching in Amharic this enables customers locate more pertinent and correct material.

People use the CRF algorithm a lot because it is very good at figuring out how words next to each other in a sentence relate to each other it forecasts chunk labels with great accuracy and picks up patterns from enormous datasets. CRF focuses on context thus it considers the words before and after before determining the chunk type. CRF often works really well in text chunking applications as language is highly context dependent.

The MBL method operates differently. MBL remembers examples it has previously encountered during training instead of studying mathematical patterns such CRF when a fresh statement shows up it matches the new words with saved examples and projects the most like chunk tag. Popular MBL

techniques are IB1 and IGTree algorithms particularly for smaller datasets MBL is simpler and more practical to implement.

Chunking Amharic language helps computers grasp the language more effectively. It improves search engines, voice recognition, grammar correction, question answering, translation systems and numerous other language tools. Using machine learning techniques like CRF and MBL helps to create more intelligent systems that help Amharic speakers and help Ethiopian language technology to advance.

4.7 Prototype of the ATC

This illustration shows how a computer translates and arranges an Amharic sentence into significant word groups. This method is known in Natural Language Processing (NLP) as shallow parsing or text chunking. The computer learns which words go together as phrases like noun phrases or verb phrases instead of sentence by sentence without meaning. The example's sentence "**ይህ አዲስ የአማርኛ ጽሑፍ ማቀናበሪያ ሥርዓት ነው**" translates as "This is a new Amharic text processing system." First the system splits the sentence into discrete words. Each word then gets a Part-of-Speech (POS) tag which informs the computer whether the word is a noun, verb, adjective, determiner or punctuation mark. For instance "**ይህ**" is categorized as a determiner meaning "this" "**አዲስ**" as an adjective meaning "new" and "**ነው**" as a verb meaning "is" following POS tagging the system uses BIO notation to chunk. Beginning inside and outside is what BIO means. These tags enable the computer to grasp where a phrase starts and ends. "B" denotes the start of a phrase; "I" indicates the word extends the phrase and "O" means the word does not fit any phrase in the provided example "**ይህ**" is categorized as B-NP which means it initiates a noun phrase. Because it starts an adverbial phrase "**አዲስ**" is marked B-ADJP. The words "**የአማርኛ ጽሑፍ ማቀናበሪያ ሥርዓት**" are grouped together as one large noun phrase because all of them describe the same object "Amharic text processing system" At last "**ነው**" introduces a verb phrase since it shows the state or action of the sentence. The letter "O" is used for the punctuation mark since it outside any phrase structure.

These predictions are made using machine learning techniques like Memory-Based Learning (MBL) and Conditional Random Fields (CRF). CRF examines the link between words and their environment. Before deciding it looks at past words, future words and prior chunk labels in addition to the present one. This enables CRF to grasp long and difficult Amharic sentence patterns more precisely. Since Amharic has a lot of word forms and rich morphology MBL resembles more human memory while CRF does rather nicely in spotting connected phrases. It keeps numerous instances from training data rather than creates a mathematical model. MBL evaluates a fresh sentence against past instances and forecasts the chunk label based on the most like ones. Because this technique recalls precise instances

from the past it is especially helpful for dealing with odd words, exceptions or erratic sentence structures.

MBL aims to remember comparable examples and apply them as needed while CRF seeks to mathematically learn linguistic rules and sentence patterns. For sophisticated sentence forms CRF usually provides better accuracy MBL is simpler to grasp and works nicely with smaller datasets or erratic language patterns. These algorithms enable computers to interpret Amharic phrases more spontaneously and power programs including machine translation, grammar correction, speech recognition and question answering and information extraction.

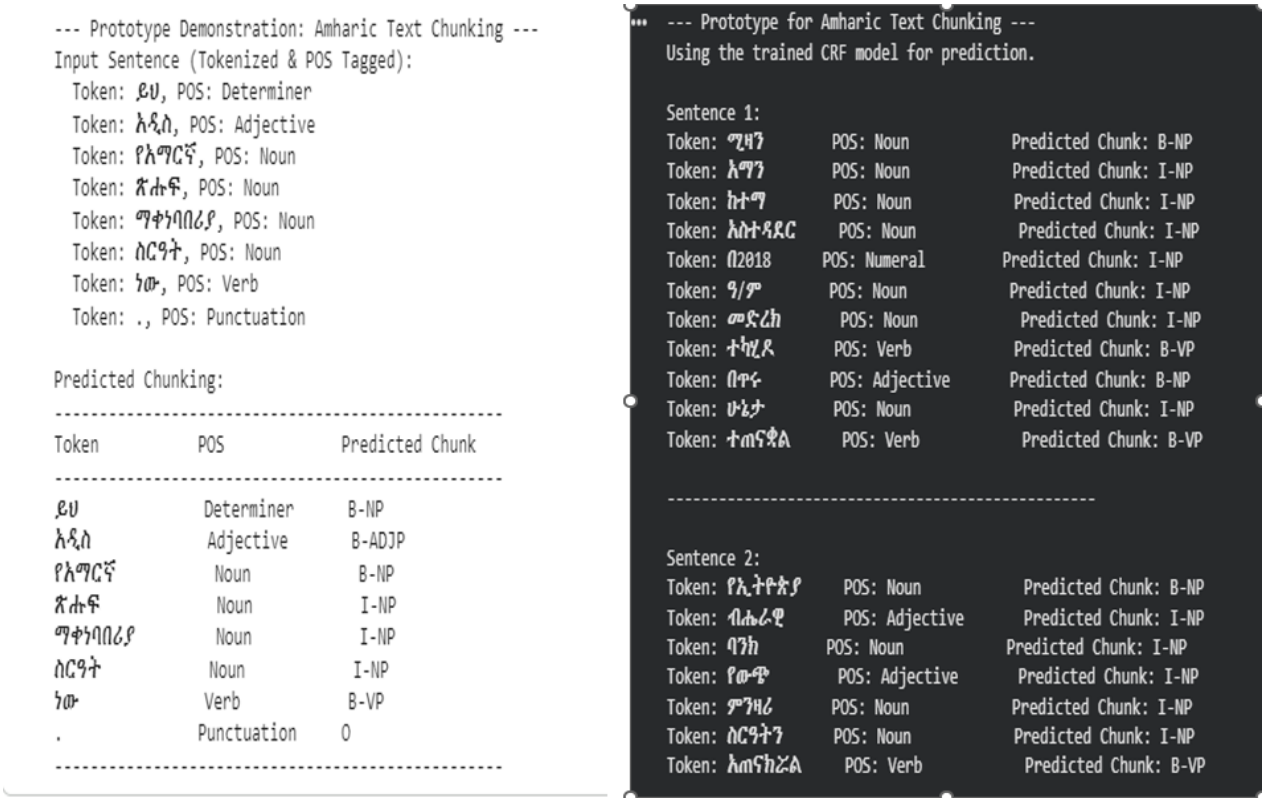


Figure 4.5 Prototype of ATC

CHAPTER FIVE

5. EXPERIMENT AND RESULT DISCUSSION

5.1 Introduction

In chapter four the design of Amharic text chunker with its implementation are discussed and this chapter going to present the process of the experiment conducted. Experiments are done in different scenarios so that the results are displayed accordingly with their explanations.

5.2 Feature Representation and Description

Feature selection plays a crucial role in CRFs framework. Experiments were carried out to find out most suitable features for Amharic text chunking task. The main features for text chunking task have been identified based on the different possible combination of available word and tag context. In this study it has been considered a different combination from the following set for inspecting the best feature set for the Amharic text chunk task. The various features used for developing our system are described below:

Surrounding word: - Preceding and following words of a current word can be used as a feature because surrounding words influence the current word and considered different window size of the word until to get a promising result.

Combination of word: - combinations such as the preceding or current word and current word or following word are used as feature.

Window for POS of the current word: - Preceding and following POSs of a present word are used as feature because surrounding POSs influence the present word.

Combination of POSs: - combination such as the preceding current POS and current following POS are used as feature.

The empirical study found that the following combination of features gives the optimal feature set.

F (best) = [$W_{i-2}W_{i-1}W_iW_{i+l}W_{i+2}$, $POSi\text{-}2POS_{i-l}POS_iPOS_{i+1}POS_{i+2}$]

5.3 Experiment in CRF-Based Amharic Text Chunking

Experiment in CRF-Based Amharic Text Chunking Statistical sequence modeling method Conditional Random Fields (CRF) which takes into account the relationships between nearby words and their grammatical functions to successfully manage natural language processing chores including text segmentation. CRF was applied in this study to create an Amharic text chunk that learns patterns from training data and forecasts chunk tags for fresh phrases based on data-extracted characteristics as well as context. Learning correlations between features and chunk tags across sequences instead of treating each word separately by integrating both word context and POS tags helps CRF to CRF may identify structural patterns in Amharic phrases manage changes in sentence form and show greater

generalization to undiscovered data. Though the advantages in accuracy shown the need of feature selection the number of extracted features and processing time grew with context size and feature richness usually CRF-based Amharic text chunking reveals a major increase in accuracy when the model takes grammatical knowledge and word context into account while employing little context windows on its own give's minimal performance combining more context with POS tags yields the most trustworthy and human-like chunking outcomes. CRFs hence provide a flexible data-driven method ideal for managing Amharic text's intricacy and inconsistency.

- CRF-Based Studies looked at several feature sets and context windows to see how they affected performance included among the characteristics were adjacent words (wi-1, wi, wi+1), larger context windows (wi-2 to wi+2) and part-of-speech (POS) tags. The model found several elements for each experiment therefore capturing the possible combinations of grammatical information and words in the CRF experiments. Every test was created around a particular set of characteristics the model employed to grasp the context of words in a sentence.
- The column Features utilized shows which inputs were taken into account as the current word, adjacent words or POS tags these aid the model to grab grammatical as well as lexical information.
- The Extracted Features column displays the richness and complexity of the knowledge the model could learn from by indicating how many unique data points or combinations were produced from the dataset based on these inputs.
- The Time Taken records capture the time the model needed to train on the dataset and process these attributes therefore offering insight on the calculation load. None the accuracy shows how well the model appropriately forecast the chunk tags for fresh phrases therefore highlighting the usefulness of each feature set. These measures taken together offer a full view of how the CRF text chunking model's performance and efficiency are influenced by the selection of features and dataset size.

5.3.1 CRF-Based Experiment

The table shows how well the Amharic text chunking model would work with different mixes of contextual word features and Part-of-Speech (POS) tags the experiments aimed to determine how various extracted features affect the model's capacity to recognize Amharic sentence chunk borders.
For every set of characteristics the outcomes cover training time, precision, F1-score and accuracy.
The model in the first experiment only considered the words immediately around it like the one before (wi-1), the one now (wi) and the one after (wi+1). The extracted features were showed as wi-1=wi is "በሚዛን" and wi+1 is "አማን" with a training time of 0.40 seconds the model reached 65% precision,

65% F1-score and 66% accuracy using this set of characteristics. This outcome shows that the model's reliance just on lexical data limits its performance even if adjacent words offer helpful contextual information for chunk detection. The technology has restricted capacity to grasp Amharic phrase syntactic links among words without grammatical elements.

Table 5.1 CRF-Based Experiment

Features Used	Extracted Features	Training Time (s)	Precision	F1-score	Accuracy
wi-1, wi, wi+1	wi-1=<START>, wi=በሚዛን, wi+1=አማን	0.40	0.65	0.65	0.66
wi-1, wi, wi+1 + POSi-1, POSi, POSi+1	wi-1=<START>, wi=በሚዛን, wi+1= አማን, POSi-1=<START>, POSi=NP	0.43.	0.84	0.83	0.84
wi-2, wi-1, wi+1, wi+2	wi-2=<START>, wi 1=<START>, wi+1=አማን, wi+2=ከተማ	0.48	0.57	0.55	0.58
wi-2, wi-1, wi, wi+1, wi+2 + POSi-2, POSi-1, POSi, POSi+1, POSi+2	wi-2=<START>, wi-1= <START>, wi=በሚዛን, wi+1=አማን, wi+2=ከተማ	0.61	0.84	0.83	0.84

The same contextual terms were supplemented with POS-tag data in the second experiment. Among the linguistic and grammatical data retrieved were wi-1 = wi is "**በሚዛን**", wi+1 is "**አማን**", POSi-1 and POSi is "Noun Phrase". This study greatly enhanced the model's performance with a training time of 0.43 seconds the system reached 84% precision, 83% F1-score and 84% accuracy. The outcome demonstrates unequivocally that POS tags give the model crucial grammatical hints that enable it to detect sentence structures more precisely. Including POS data helped the model to better identify verb phrases and noun phrases, therefore defining chunk borders. The training time only changed a little even with extra elements added. In the third experiment, the contextual window was enlarged using two prior words and two next ones free from current word or POS tags. Among the retrieved characteristics were wi-2= wi-1= wi+1 is "**አማን**" and wi+2 = "**ከተማ**". The model performance dropped even with the bigger framework window. With a training period of 0.48 seconds, the test reached 57% precision, 55% F1-score, and 58% accuracy. This outcome implies that raising the quantity of surrounding words devoid of grammatical information could introduce extraneous or loud data into the model. The classifier consequently becomes less good at precisely marking chunk borders.

The fourth study mixed POS-tag data with more general contextual words. Together with their related POS tags, the retrieved characteristics comprised wi-2 = wi-1 = wi is "**በሚዛን**", wi+1 is "**አማን**", wi+2 is "**ከተማ**." with 84% precision, 83% F1-score and 84% accuracy this experiment showed the highest

overall performance. The training time of 0.61 seconds was marginally more than earlier studies since more features were handled still the performance improvement shows that by combining grammatical features with more general contextual information, the model can grasp sentence structure better and more precisely mark chunk limits.

Generally the test show that improving Amharic text chunking performance depends much on POS-tag characteristics models that blended POS tags with contextual terms always had far higher precision, F1-score and accuracy than models that only drew on lexical context. The findings also reveal that just lengthening the contextual window devoid of grammatical data might even lower categorization accuracy and does not always enhance performance therefore the most efficient feature representation for Amharic text chunking results from combining lexical characteristics with POS-tag data

5.4 Memory-Based Text Chunk Implementation

The memory-based method combines the best of both knowledge-based and probabilistic techniques to address NLP problems they categorize by analogy the method identifies comparable cases in memory when a new instance arises and forecasts its class based on these parallels. To carry out memory based on Amharic material. TiMBL (Tilburg Memory-Based Learner) uses chunking as a teaching tool. TiMBL lets one keep examples in memory and categorize fresh cases according to their degree of similarity to past ones. Classifiers and parameters of model TiMBL memory-based learning calls for close adjustment of settings for our tests.

- Distance Metric: DMV (Modified Value Difference Metric)
- Feature weighting: No weighting
- k in nearest neighbor: -Default (optimized for accuracy)

Two classifier engines tested it:

- IGTree: Constructs a decision-tree-like structure in memory and categorizes it depending on feature relevance (Gain Ratio) with these classifiers it builds models for later use in chunking.
- IB1: Categorizes by matching the closest stored examples to a new instance.

Text chunking based on memory operates quite organically. The system just remembers cases instead of attempting to learn challenging rules or formulas. It searches back at comparable situations it has already seen and draws on them to decide when it comes across a fresh word or phrase in essence it learns by means of comparison and connection of fresh data to experiences much as humans learn language. This strategy is successful in managing frequently sophisticated and irregular natural language since it blends the advantages of knowledge-based techniques with data-driven ones.

Using TiMBL (Tilburg Memory-Based Learner) the model was applied. TiMBL looks for the closest match among the previously stored training examples when a fresh input arrives by comparing it with them based on its similarity. It projects the appropriate chunk tag. This allows the system to be flexible

as it changes based on the data it has seen to make the model operate well rather than depending on set rules. It fixed a number of crucial factors. Using the Modified Value Difference Metric it measured the degree of similarity among several characteristics this is especially helpful for dealing with categorical words and POS tags. Additionally deciding against using feature weighting meant that all characteristics were treated equally. Also it used the default k value in the nearest neighbor method which let the system automatically decide how many similar instances to take into account while making a decision.

Two other classifier kinds IB1 and IGTree are tried here. The IB1 classifier selects the most similar stored example by directly comparing a new instance with all of them. Because it uses minute comparisons this makes it simple and sometimes more accurate conversely IGTree arranges the data into a tree-like form and uses a technique called Gain Ratio to determine which attributes are more crucial. Although this lessens the number of comparisons required it occasionally misses finer details that somewhat lower accuracy this accelerates it learning based on memory behaves in a manner somewhat like human thought. It uses past experiences as a foundation for decisions by matching them with fresh circumstances this hence fits naturally and efficiently for tasks including text chunking.

5.4.1 Experiment with IB1

The IB MBL approach for Amharic text chunking based on several grammatical and contextual elements. The goal of the study was to assess how different feature combinations influence the detection of syntactic chunks in Amharic sentences an NLP job called text chunking group words into significant phrases like noun phrases (NP) or verb phrases (VP) choosing appropriate elements is very crucial for enhancing chunking performance given that Amharic is a morphologically rich language with sophisticated sentence forms.

All tests use IB1 a memory-based learning strategy. IB1 uses comparison with past training examples to categorize a new word and allocate the chunk label according to similarity. The Modified Value Difference Metric (MVDM) appropriate for managing categorical characteristics like words and POS tags, was used to assess the similarity among examples. No weighting was used in any of the tests; therefore every characteristic helped equally during classification. The IB1 machine learning model's performance on Amharic text chunking utilizing various mixes of contextual word and Part-of-Speech (POS) characteristics is shown below. Every experiment sought to see how various feature sets affect the model's capacity to accurately detect Amharic sentence chunk limits.

The first experiment model only considered the immediately surrounding words as characteristics: the word before, the current word, and the following one with precision, recall and F1-score values near 60% this study attained an accuracy of roughly 60% with few attributes the model trained really fast the training time was just 0.11 seconds. But the performance was still somewhat poor as the model

relied just on lexical context without any grammatical information. This outcome implies that Amharic text's chunk borders cannot be precisely determined by surrounding words alone. Because Amharic words have sophisticated morphological changes, the model could find it difficult to identify phrase structures just on adjacent words.

Table 5.2 IB1-Based Experiment

Features Used	Algorithm	Distance Metrics	Feature weighting	Training Time (s)	Precision	F1 score	Accuracy
wi-1, wi, wi+1	**IB1**	**MVDM**	**No weighting**	**0.11**	**0.59**	**0.60**	**0.60**
wi-1, wi, wi+1 & POSi-1, POSi, POSi+1	**IB1**	**MVDM**	**No weighting**	**0.13**	**0.79**	**0.79**	**0.79**
wi-2, wi-1, wi, wi+1, wi+2	**IB1**	**MVDM**	**No weighting**	**0.18**	**0.57**	**0.57**	**0.57**
wi-2, wi-1, wi, wi+1, wi+2 + POSi-2, POSi-1, POSi, POSi+1, POSi+2	**IB1**	**MVDM**	**No weighting**	**0.23**	**0.78**	**0.78**	**0.78**

Adding Parts-of-Speech (POS) tags to the contextual words helped to expand the feature set of the second experiment. The model took both grammatical aspects (POSi-1, POSi, POSi+1) and lexical elements (wi-1, wi, wi+1). The system's performance was greatly enhanced by this. Precision and F1-score all rose significantly along with the 79% accuracy though the performance gain was significant, the training time rose somewhat to 0.13 seconds this outcome shows that POS tags give crucial syntactic data that enables the classifier to better spot phrase patterns in Amharic sentences. For instance, noun phrases and verb phrases are closely related with particular grammatical classifications, and POS tags enable the model to more precisely identify these structures. The study unequivocally demonstrates that Amharic chunk more efficient feature representation results from lexical and grammatical data run together.

Without employing POS tags the third experiment's context window was broadened to cover two prior and two following words (wi-2, wi-1, wi, wi+1, wi+2) the model performance actually dropped to only around 57% accuracy though more appropriate terms were added, the lack of grammatical information limited the model's ability to generalize well. The larger background might have included useless or loud information that complicates the classification process the training time also went up to 0.18 seconds as more features called for more comparisons during learning. This study shows that merely raising the number of environment words does not always enhance chunking ability too much lexical context without syntactic guidance might sometimes perplex the classifier and lower accuracy.

The Last Experiment's model combined POS tags with the larger contextual window the feature set comprised adjacent words and their related grammatical tags wi-2, wi-1, wi, wi+1, wi+2 along with

POSi-2, POSi-1, POSi, POSi+1 and POSi+2 the model's roughly 78% accuracy is far better than those of tests employing just words. Precision and F1-score values also came in at around 78% the outcome though was somewhat less than the second experiment when just immediate neighboring words with POS tags were utilized using a very wide context window could thereby introduce unneeded or superfluous information that somewhat compromises generalization performance. The model's processing of more elements during training helped to further raise the training time to 0.23 seconds. Though the larger background gave more data the simpler feature set with immediate context and POS tags generally fared better.

The experimental findings show generally that POS tags significantly help to enhance Amharic text chunking quality while the addition of grammatical elements much improved chunk identification models depending just on contextual cues generated less precision. The best outcome was close to 80% accuracy using straight adjacent words combined with their parts of speech. This suggests that Amharic text phrase boundary identification is greatly aided by syntactical knowledge together with local contextual data the study also finds that expanding the context window does not always result in improved results. More features could make calculations more difficult and provide unneeded information particularly if there is not enough grammatical direction therefore using memory-based learning techniques like IB1 choosing a good feature combination is critical for developing a quick and accurate Amharic chunking system.

5.4.2 Experiments with IGTree

The IGTree classifier was applied to a much larger dataset containing sentences rather than the smaller datasets used in earlier tests the purpose of scaling up the dataset was to test the model’s performance on a more realistic extensive corpus and see how well it can handle more diverse sentence structures and vocabulary the setup remained consistent with previous IGTree experiments.

- Classifier: IGTree which organizes training examples into a tree-like structure this makes classification efficient because the model does not need to compare a new instance with every single example in memory instead it traverses the tree using the most informative features first.
- Distance metric: Overlap which checks whether feature values match exactly this is straightforward but effective for categorical data like words and POS tags.
- Feature weighting: Gain Ratio which determines which features are most informative for predicting chunk tags. Features that provide more information about the structure of a sentence are given priority in the decision process.
- By applying this setup to a sentence dataset the model now has access to a far larger and more varied set of examples this means it can learn more patterns, handle different types of phrases and make predictions that are more generalized. The larger dataset also allows the IGTree model to

build a more robust tree since it has more examples to evaluate feature importance and refine decision path.

The table shows below the results of experiments with the IGTree machine learning technique on Amharic text chunking using different mixes of Part-of-Speech (POS) tags and contextual word features these studies sought to evaluate how well the model could correctly identify chunk boundaries in Amharic sentences using varied feature representations. In Natural Language Processing (NLP) text chunking is a very important job since it arranges words into meaningful syntactic units like noun phrases, verb phrases and prepositional phrases. Given the structural complexity and morphological richness of the Amharic language choosing appropriate characteristics is essential to improve chunking performance.

The IGTree approach was used throughout the tests together with the Overlap distance measure and the Gain Ratio for attribute weighting the Overlap metric evaluates similarity by seeing if the instance's feature values match or differ one technique of feature selection meant to help the model find the most important features for categorization is Gain Ratio emphasizing important elements helps Gain Ratio improve categorization assignment learning speed as well as accuracy.

Table 5.3 IGTRee Experiment

Features Used	Algorithm	Distance Metrics	Feature weighting	Training Time (s)	Precision	F1 score	Accuracy
wi-1, wi, wi+1	IGTRee	Overlap	Gain Ratio	929.08	0.61	0.59	0.61
wi-1, wi, wi+1 & POSi-1, POSi, POSi+1	IGTRee	Overlap	Gain Ratio	34.16	0.85	0.84	0.85
wi-2, wi-1, wi, wi+1, wi+2	IGTRee	Overlap	Gain Ratio	1451.87	0.59	0.58	0.60
wi-2, wi-1, wi, wi+1, wi+2 +POSi -2, POSi-1, POSi, POSi+1, POSi+2	IGTRee	Overlap	Gain Ratio	48.72	0.85	0.84	0.85

The first experiment only used the three immediate contextual words the one before (wi-1), the one now (wi) and the one after (wi+1) as features with this set of attributes the model reached a precision of 61%, F1-score of 59% and accuracy of 61% still the training process took a long time around 929.08 seconds. These results show that just using neighboring words produces little contextual knowledge for spotting Amharic text chunk borders. The lack of grammatical knowledge limits the model's capacity to grasp the syntactic connections among words. Thus even after the extended training period the model's chunking ability was normal. This study shows how Amharic sentence structure cannot be correctly portrayed using only lexical context.

In the second experiment grammatical insights were added together with the same contextual words using POS tags. The feature set included the following last word, current word, next word and their

corresponding POS tags (POSi-1, POSi, POSi+1) this improvement in performance was mostly brought about by this adjustment. The model reported precision of 85%, F1-score of 84% and accuracy of 85% at the same time the training time was significantly cut from 929.08 seconds to only 34.16 seconds. This outcome emphasizes the importance of POS tag data in relation Amharic text chunking. Essential grammatical and syntactical cues found in POS tags help the model to more precisely detect phrase structures for instance verb phrases relate with verbs and auxiliaries whereas noun phrases usually associate with nouns and adjectives. POS tags helped the IGTree model to more precisely and efficiently define chunk boundaries. The reduced training period also implies that during the learning phase grammatical elements help the model to better organize decision constructs.

Including two past words and two following words (wi-2, wi-1, wi, wi+1, wi+2) without employing POS tags helped the third study to broaden the contextual window. Though the larger background was there the model performance dropped. The test obtained 59% precision, 58% F1-score and 60% accuracy. Furthermore the training time shot up to 1451.87 seconds making it the most among all trials. This outcome implies that just raising the number of contextual words does not always enhance chunking execution. Without grammatical direction from POS tags the other words around could bring unnecessary or noisy information into the model. The bigger feature set also raises computational complexity, therefore slowing down and less efficient training process thus only broader lexical context is insufficient to raise Amharic chunk classification.

In the fourth test POS-tag data were combined with broader contextual terms. Two past words, the present word, two following ones and their associated POS tags were all taken into consideration by the model. This experiment had the best overall results 85% precision, 84% F1-score and 85% accuracy. Reducing the training time to 48.72 seconds far less than the tests depending just on contextual words the outcome shows that the IGTree model may better grasp Amharic sentence structure and more precisely identify chunk borders when grammatical features are added with lexical information. POS tag inclusion helped the model use the broader context properly free from noise or extraneous information.

The tests generally demonstrate that Amharic text chunking performance using the IGTree method depends very much on POS-tag features. Compared to models depending only on word context models employing both contextual words and POS tags always had significantly higher precision, F1-score and accuracy. The studies also show that expanding the context window free grammatical information might raise complexity, slow down training and lower performance thus the most useful feature representation for Amharic text chunking employing the IGTree machine learning model comes from the combination of contextual lexical elements and POS-tag data.

5.5 Dataset Distribution Analysis

Dataset Distribution Analysis is the process of examining how data is spread across different classes, values or categories in a dataset to understand balance, imbalance and overall structure before modeling

5.5.1 Pre-SMOTE distribution of IOB2 chunk labels in ATC dataset

The following image shows the class distribution of the dataset prior to the application of any balancing approach the number of tokens that belong to each IOB2 chunk label utilized in the Amharic text chunking activity is displayed the output is given in a simple console format the dataset shows 11 distinct chunk labels with extremely uneven token distributions with 2,890 and 2,093 occurrences respectively Noun Phrase labels like I-NP and B-NP have the most tokens. B-VP and I-VP are common examples of verb phrase labels however certain classes like I-ADJP and I-ADVP only have 11 and 6 tokens respectively making them very sparse. This shows that the dataset is significantly imbalanced with certain classes dominating and others being underrepresented two status messages demonstrating the execution of the SMOTE (Synthetic Minority Over-sampling Technique) procedure are displayed in the console output after the class distribution. The second message confirms that the process was successfully completed whereas the first message states that SMOTE is being applied and may take some time in general the image illustrates the presence of substantial class imbalance in the dataset prior to balancing. A significant portion of chunk classes particularly noun phrase labels have a higher number of samples than minority classes such as adjective phrase and adverb phrase labels. Because the model may develop a bias toward the majority classes this disparity might have a detrimental impact on machine learning model performance. SMOTE is used to produce more samples for the minority classes and better balance the dataset.

```
Class distribution BEFORE SMOTE (Token-level):
'I-NP': 2890
'B-NP': 2093
'B-VP': 1301
'I-VP': 579
'B-PP': 321
'O': 116
'B-ADJP': 102
'B-ADVP': 89
'I-PP': 20
'I-ADJP': 11
'I-ADVP': 6

Applying SMOTE... This might take a moment.
SMOTE applied successfully.
```

Figure 5.1 Class distribution before SMOTE (Token - Level).

The distribution of chunk tags in the dataset prior to SMOTE is shown in the bar chart. The distinction between the different IOB2 chunks tags employed in the Amharic text chunking exercise is evident from the chart the number of tokens associated with each chunk tag is shown on the y-axis while the

numerous chunk tags are shown on the x-axis. For improved readability the bars are shown with a gradient color scheme ranging from green and yellow tones to dark purple and blue hues the chunk labels are somewhat spun according to the chart certain chunk types contain a lot of tokens whereas others only have a small number the I-NP tag is followed by the B-NP which has the highest frequency and is shown as the tallest bar in the graph. These two markers are prevalent throughout the dataset. A fair quantity of tokens are present in other courses like I-VP and B-VP the little bars representing the minority groups such as I-ADVP, I-ADJP and I-PP are nearly indiscernible around the baseline.

The graphic indicates a substantial class imbalance in the original dataset prior to balancing. Although there may be thousands of tokens in majority classes particularly noun phrase tags only a few cases may be found in minority classes. Since they may not learn effectively and may prioritize the majority classes over the minority ones such imbalance could have a negative impact on machine learning models. Thus, there is compelling visual proof in the graph that SMOTE should be used to balance the dataset by raising the sample size in the underrepresented groups.

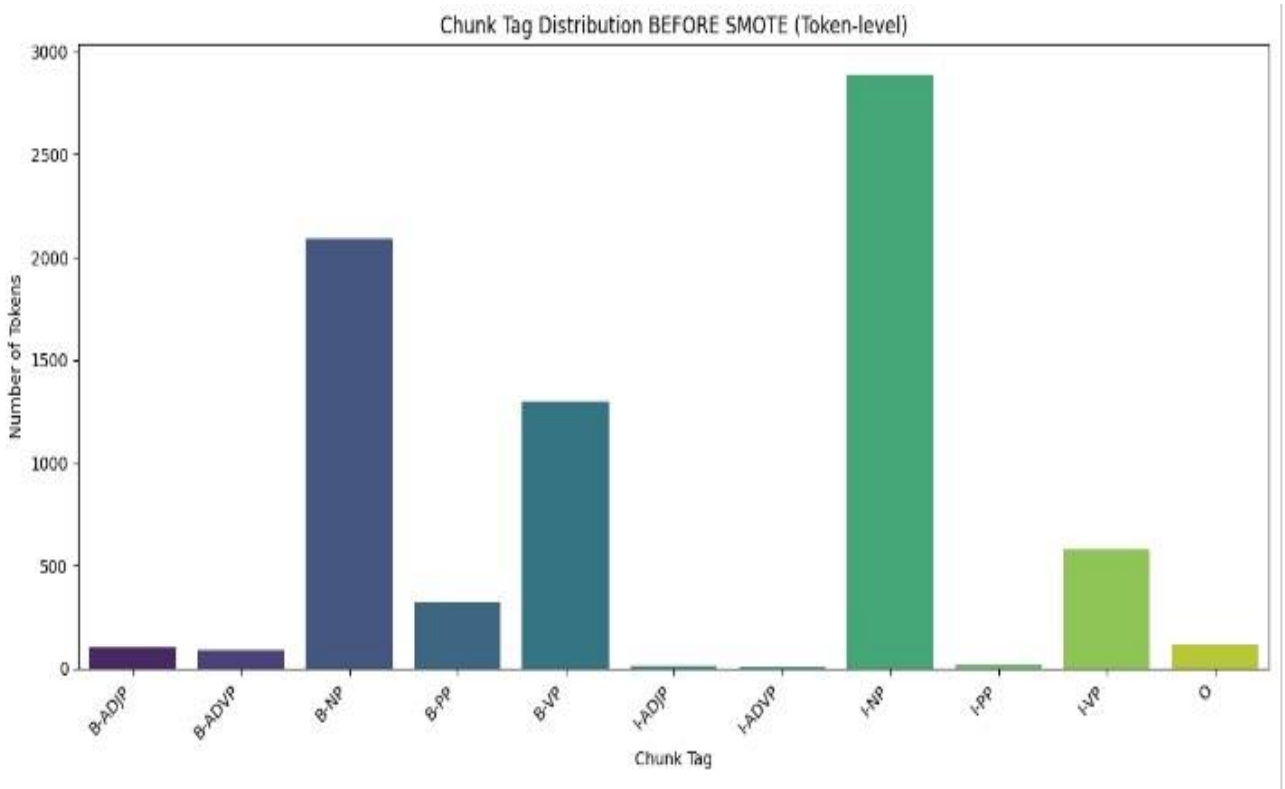


Figure 5. 2 Chunk tag distribution before SMOTE (Token - Level)

5.5.2 Balanced Class Distribution AFTER SMOTE in ATC Dataset

The class distribution of the dataset following the application of SMOTE (Synthetic Minority Over-sampling Technique) at the token level is shown in the image. It is shown in a straightforward console output style under the Class distribution AFTER SMOTE (Token-level) heading. After balance the

output shows all the IOB2 chunk labels that were used in the chunking task for the Amharic text along with their token counts.

The dataset consists of eleven chunk categories such as adverbial phrases, verb phrases, noun phrases, adjective phrases, prepositional phrases and the "O" outside tag. According to the image each class now contains precisely the same amount of tokens 2890. SMOTE made the dataset fully balanced.

Before balancing some chunk labels such as noun phrases, had thousands of examples but others such as adjective and adverb phrases had very few due to this disproportion the machine learning model may be prejudiced against majority classes and underperform for minority classes. Synthetic samples were created for the smaller categories following the application of SMOTE until all of the chunk tags were the same size.

The picture unmistakably shows the success of the balancing procedure in order to give them equal representation with big categories like B-NP and I-NP minority classes like I-ADVP, I-ADJP and I-PP were substantially enhanced. This fair distribution aids the machine learning model in learning all chunk categories more equitably and successfully overall the picture shows that SMOTE effectively addressed the class imbalance issue in the Amharic text chunking dataset. It is anticipated that the balanced dataset will enhance the model's ability to learn, lessen prediction bias and boost the overall performance of all chunk labels in terms of categorization.

```
Class distribution AFTER SMOTE (Token-level):
'B-PP': 2890
'B-VP': 2890
'B-NP': 2890
'I-NP': 2890
'I-VP': 2890
'B-ADJP': 2890
'I-ADJP': 2890
'B-ADVP': 2890
'O': 2890
'I-PP': 2890
'I-ADVP': 2890
```

Figure 5.3 Class distribution AFTER SMOTE (Token - Level)

The graph illustrates which describes the distribution of chunk tags in the dataset after applying the SMOTE balancing approach on the horizontal axis of this chart a variety of IOB2 chunk tags are shown including B-ADVP, I-ADJP, I-NP, B-PP, I-VP, B-VP and B-NP among others the amount of tokens associated with each tag is represented by the vertical axis all of the bars are almost the same height which is immediately apparent. After applying SMOTE each chunk tag has about the same amount of tokens between 2,800 and 2,900 each meaning that they are all roughly equally distributed to put it another way the entire dataset has been balanced.

Some tags would typically appear much more often before SMOTE and unusual tags would have significantly less examples but now even those previously uncommon classes appear as often as the major classes like B-NP or I-NP the dataset is no longer controlled by a single class and none are underrepresented a fluid gradient that goes from deeper purples and blues on one side to lighter pink, orange and yellow hues on the other colors the bars this color shift is purely for visual aesthetics it doesn't signify any value difference since all bars have the same number of counts.

In conclusion the chart plainly illustrates the effects of SMOTE the dataset has been made uniform throughout all chunk tags since it minimizes bias toward common labels and enhances the model's capacity to properly predict all chunk types this equilibrium is crucial for ensuring that machine learning models treat all classes equally during training.

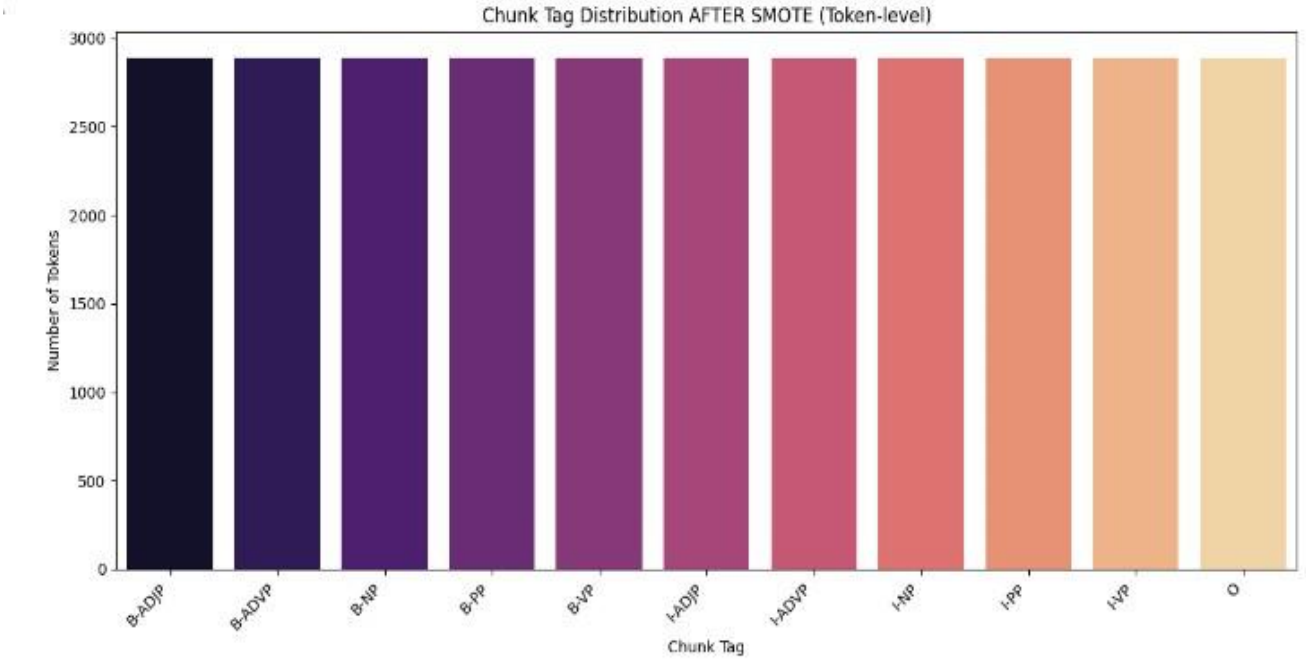


Figure 5.4 Chunk tag distribution AFTER SMOTE (Token - Level)

5.6 Baseline Study ATC Models

This study evaluates how effectively the suggested Amharic text chunking technique works using baseline models as reference systems these models assist us to evaluate the fresh CRF and MBL-based method with other current machine learning methods to find if it truly beats them.

The baseline models comprise several often-used methods in natural language processing including Hidden Markov Models (HMM), Support Vector Machines (SVM) and Memory-Based Learning techniques such IB1 and IGTree these techniques are frequently employed for chores such sequence labeling and text chunking; hence they provide a reasonable and significant approach of performance testing.

5.6.1 Hidden Markov Model (HMM)

One of the earliest techniques for sequence work like text chunking is Hidden Markov Model (HMM). It was also applied in earlier Amharic language processing research occasionally alongside rule-based techniques to identify chunk boundaries simply put HMM seeks to predict the function of every word in a sentence (say noun phrase or verb phrase) depending on patterns it discovered from training data. It mostly verifies two things which words are likely to go with some labels and what label usually precedes another the fundamental idea is that every word's label relies mostly on the one before it. Though it also means the model cannot totally grasp lengthy or complex sentence relationships this simplifies and speeds the model consequently HMM often only offers passable results particularly for languages like Amharic with complicated word structures and little training data. It is used as a basic conventional model for comparison in this investigation.

5.6.2 Support Vector Machine (SVM)

Classification uses the machine learning technique known as Support Vector Machine (SVM) it helps to define a sharp line between several data sets SVM considers aspects like surrounding words, part-of- speech tags and contextual data for chunking to ascertain the right label for every word unlike sequence models SVM does not naturally know word order it still performs really well in classification problems though particularly when strong features are created.

Its efficiency is mostly determined by the preparation of the traits here SVM helps one to contrast a robust conventional classification technique with the suggested CRF-based sequence model.

5.6.3 Learning Based on Memory (IB1)

MBL approaches like IB1 function quite easily rather than memorizing difficult rules they keep all of their training data and apply it straight forwardly in their predictions.

IB1 finds the most similar past examples to a new word and uses them to guess its chunk label for Amharic text it contrasts the context of the present word with related examples from the training data. This approach is simple to use and does not call for a lot of training but it can become sluggish if the dataset is big its accuracy relies on how similarity is evaluated.

5.6.4 Memory-Based Learning (IGTree)

An upgrade version of IB1 is IGTree it organizes the data into a tree structure according to the most important characteristics rather than looking through all the stored examples this allows the model to forecast far more quickly while maintaining good accuracy following the most useful routes in the data helps to cut pointless comparisons this study includes IGTree because it is a fast memory-based technique that has proven to be effective for several natural language processing jobs.

5.6.5 Reasons for Utilizing Baseline Models

This study comprises the baseline models for a number of straightforward yet significant purposes They enable us to contrast the suggested approach with accepted current techniques.

- They reveal if the suggested model's progress is genuine and significant they are several approaches of learning probabilistic (HMM), classification-based (SVM) and memory-based (IB1 and IGTree).
- They enable us to see how non-sequential techniques' output contrasts with sequential models like CRF' they offer a just and unbiased means of assessing the Amharic chunking system.
- Lastly the proposed model's performance is compared to that of baseline models under the same experimental settings to enable one to plainly detect the variations in performance.

5.7 Dataset Split/ Distribution

The initial dataset file has 12,000 rows of data every row corresponds to a single token a word or punctuation mark along with its related linguistic information the method however does not handle these tokens one at once. Preprocessing involved grouping the tokens using their Sentence ID into whole sentences. Following categorization the dataset provided 806 significant Amharic sentences overall. Therefore the machine learning models were trained and evaluated using these 806 complete sentences instead of the raw 12,000 token rows. From the 806 sentences in all Training the model was done with 644 sentences or 79.90% of the total the other 162 sentences or 20.10% were utilized for testing and evaluating the suggested system's performance.

Table 5.4 distribution of training and testing set.

Total sentences in dataset: 806		
Training set:	644 sentences	(79.90% of total)
Testing set:	162 sentences	(20.10% of total)

The chart visualizing the distribution of training and testing sets has been successfully displayed.

It shows the percentages of dataset allocated to each set providing a clear overview of data split.

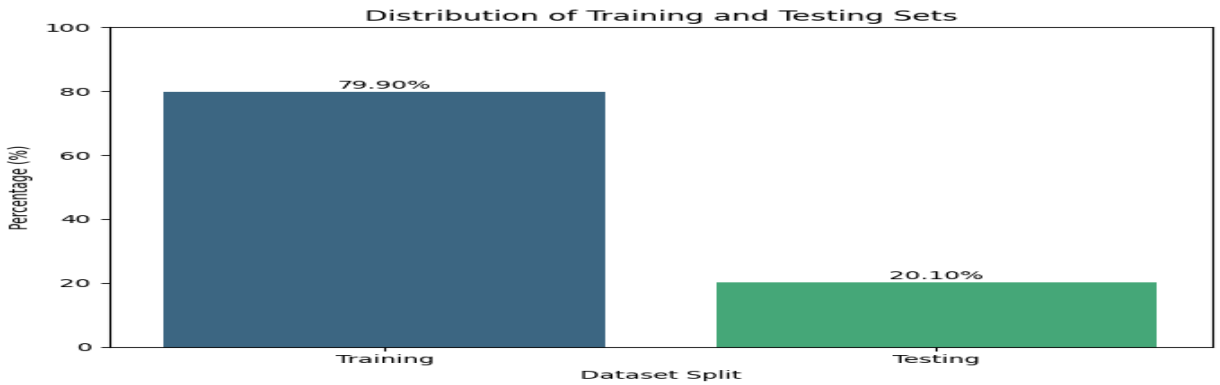


Figure 5.5 distribution of training and testing set

5.7 Proposed CRF Model Development

The Conditional Random Field (CRF) technique was used in Amharic text chunking to build the suggested model for this work a sequence labeling machine learning technique CRF helps to find relationships between close words and chunk tags inside a sentence important linguistic and contextual elements like the present word, adjacent words and part-of speech tags were taken from the Amharic text data set during model development these characteristics were then applied to train the CRF model to identify and forecast prepositional phrases, verb phrases and noun phrases that is chunk structures. The CRF model takes into account the order of words in sentences something traditional classification techniques don't do and hence produces more precise chunk predictions. The model better captures Amharic sentence grammar since it learns how chunk labels rely on surrounding words and past labels.

5.7.1 Chunk-level evaluation

The results of the chunk-level analysis indicate that the suggested Amharic text chunking model did rather well in finding and tagging sentence structures with a Precision score of 0.858 the model shows that most of the chunks the system forecasts were accurate a recall score of 0.860 means the model correctly picked up a lot of the actual chunk structures present in the dataset furthermore the F1-score of 0.852 points to an equal overall performance between Recall and Precision these results show Amharic text's significant phrase structures are well-recognized by the suggested machine learning model here is a thorough overview of the model's performance in a markdown table format with thorough metrics for every chunk type and general averages.

Table 5.5 Chunk-level evaluation (Weighted Average across all labels)

No	Metric	Value
1	Precision	0.858
2	Recall	0.860
3	F1-Score	0.852

The following chart visualizing the Chunk-level Evaluation (Weighted Average across all labels)

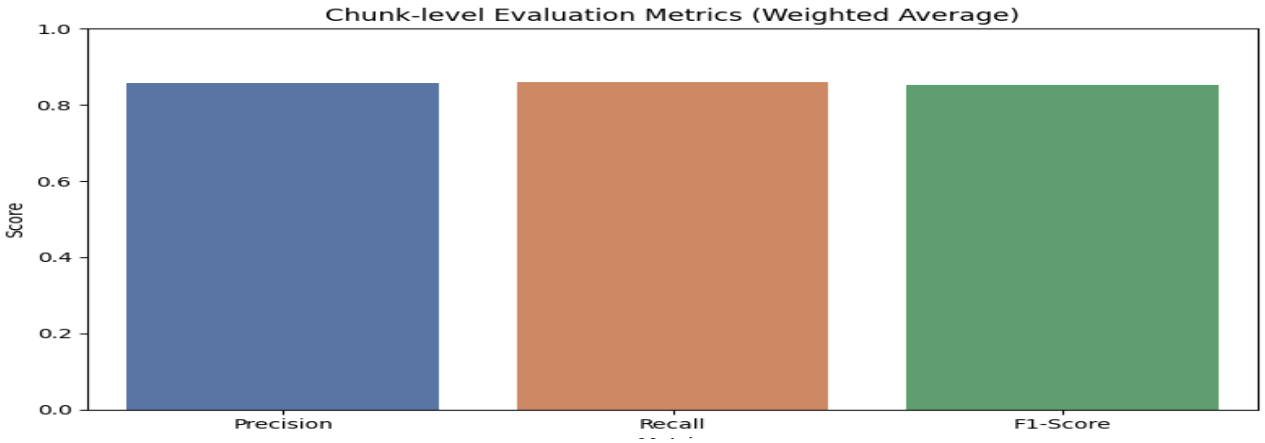


Figure 5.6 Chunk Level Evaluation Metrics (Weighted Average)

5.7.2 Token-Level Evaluation

Token-Level Assessment the token-level analysis reveals that the CRF model helps Amharic sentences to be better understood in terms of how words cluster together. The model's overall Micro F1-score of 0.860 suggests it is usually effective in assigning words to their proper chunk class's simple terms the most often used sentence patterns work particularly nicely with the model it can tell noun phrases (NP) from verb phrases (VP) very well for instance it does a respectable job identifying words within noun phrases (I-NP is 0.884) and is even more precise in finding the beginning of verb phrases (B-VP is 0.942). It also consistently picks words within verb phrases (I-VP is 0.889). This demonstrates how deftly the model has picked up the primary grammatical patterns of the language.

The model also works rather well with other structures like prepositional phrases (PP) and adverb phrases (ADVP) therefore it is able to manage more than just the fundamental sentence structures.

It does however have trouble with less frequent patterns like adjective phrases (ADJP) and some internal-phrase tags like I-PP and I-ADJP. This is essentially because the model had fewer chances to appropriately learn these since it lacked enough training samples of them in certain unusual situations like B-SBAR and I-ADVP the model could not learn anything which led in a zero score this most often arises when there is little or no training data for those labels the findings generally indicate that the model captures Amharic sentence structure well particularly for common phrase patterns but it still requires more evenly distributed data to enhance performance on uncommon or less common patterns.

Table 5.6 Token-Level Evaluation

Overall CRF Model Accuracy (Micro F1-score) is 0.860				
	Label	**Precision**	**Recall**	**F1-Score**
1	B-NP	0.870438	0.781967	0.823834
2	I-NP	0.834990	0.939597	0.884211
3	B-VP	0.938028	0.946023	0.942008
4	B-PP	0.800000	0.844444	0.821622
5	I-VP	0.906593	0.873016	0.889488
6	B-ADVP	0.687500	0.880000	0.771930
7	O	0.840000	0.677419	0.750000
8	B-ADJP	0.500000	0.152174	0.233333
9	I-PP	1.000000	0.100000	0.181818
10	I-ADJP	1.000000	0.100000	0.181818
11	B-SBAR	0.000000	0.000000	0.000000
12	I-ADVP	0.000000	0.000000	0.000000

5.8 ROC Curve Evaluation of ATC Classification Performance

The classification model's performance for various chunk tags is shown in the image Receiver Operating Characteristic (ROC) Curve (One-vs-Rest) using the One-vs-Rest (OvR) ROC analysis approach the graph assesses the model's ability to differentiate between classes by contrasting the True Positive Rate (Sensitivity) with the False Positive Rate.

The graph has two axes that span from 0 to 1. The False Positive Rate is shown on the x-axis and the True Positive Rate is displayed on the y-axis from the bottom left to the top right corner of the graph a black dashed diagonal line crosses it this line illustrates how well a random classifier performs when it has no predictive power. A high true positive rate and a low false positive rate are achieved which shows superior classification performance on any ROC curve that approaches the upper-left corner.

The chart displays several ROC curves each of which corresponds to a distinct chunk tag in the dataset. The majority of the curves are extremely close to the top-left corner suggesting that the model performs exceptionally well for the majority of classes. The graphic also shows a micro-average ROC curve with an AUC score of 0.99 demonstrating outstanding overall classification performance for all chunk labels several classes such as I-ADJP, O, I-ADVP, B-VP and B-ADVP achieve exceptionally high AUC scores that are near or equal to 1.00 suggesting almost flawless categorization performance classes such as I-NP, B-ADJP and B-PP also yield remarkable findings with high AUC values implying that the model is adept at identifying these chunk categories.

A handful of classes though exhibit poorer performance the I-PP curve exhibits a stepwise pattern rather than a smooth curve and has a lower AUC value this typically signifies a restricted number of test samples or reduced confidence levels in prediction for the class furthermore the diagonal baseline is almost precisely where the curves for B-SBAR and I-Vp are located yielding an AUC of 0.50 this demonstrates that the model was unable to extract relevant patterns for these classes and performed comparably to arbitrary speculation the label I-Vp also seems to include a formatting discrepancy or a duplication problem during the labeling procedure.

Generally the ROC visualization shows that the suggested machine learning model is highly effective for the majority of chunk tags in the Amharic text chunking task although the very high micro-average AUC value indicates that the model's overall classification ability is strong the presence of a few weaker classes suggests that additional data preparation or model refinement may be necessary.

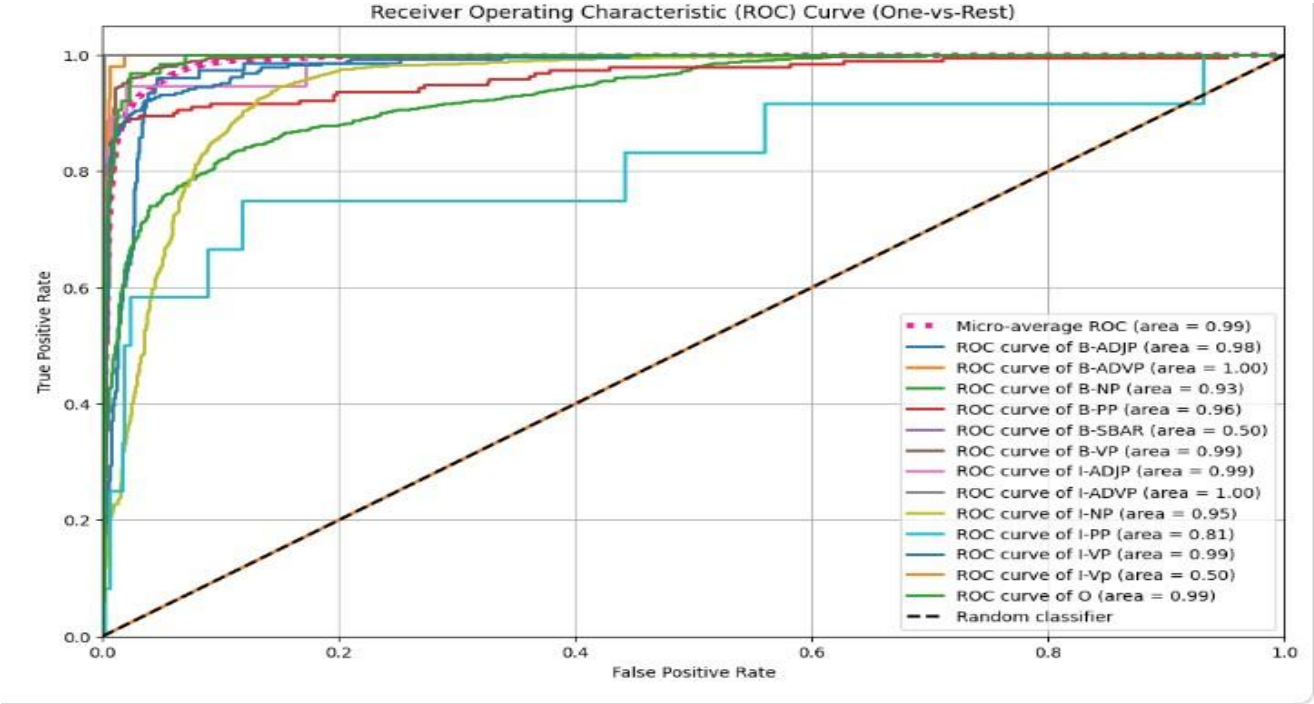


Figure 5.7 Receiver Operating characteristic (ROC)

5.9 Measures of Model Performance

Proposed Amharic text chunking system performance in prediction and categorization is measured using evaluation metrics using Accuracy, Precision, Recall and F1-score this study looks at the performance of the machine learning models. Precision counts how many expected chunks are genuinely accurate; accuracy assesses the general correctness of the chunk predictions. Recall counts how many correct chunks the model correctly detects F1-score gives the balanced performance between Precision and Recall. These performance indicators help show how well the suggested machine learning models can find and tag significant Amharic text chunk patterns.

5.9.1 Model with CRF General Performance Indicators

The CRF paradigm demonstrates great and consistent Amharic text chunking performance with an accuracy (Micro F1-score) of 0.860 it successfully predicts most chunk labels its precision (0.858) and recall (0.860) are likewise quite near to one another therefore demonstrating the model's accuracy in predictions and consistency in finding pertinent chunks with a weighted F1-score of 0.852 the model shows great general balance between precision and recall.

- The CRF model is a robust method for Amharic text chunking as it runs consistently and dependably among other models.

Table 5.7 CRF Model - Overall Metrics

1	accuracy (micro F1-score):	0.860
2	weighted precision:	0.858
3	weighted recall:	0.860
4	weighted F1-score:	0.852

The diagram shows below the overall performance metrics for the CRF model including Accuracy (Micro F1-score) weighted Precision, weighted Recall and Weighted F1-score. These scores are all above 0.85 indicating a strong overall performance for the CRF model.

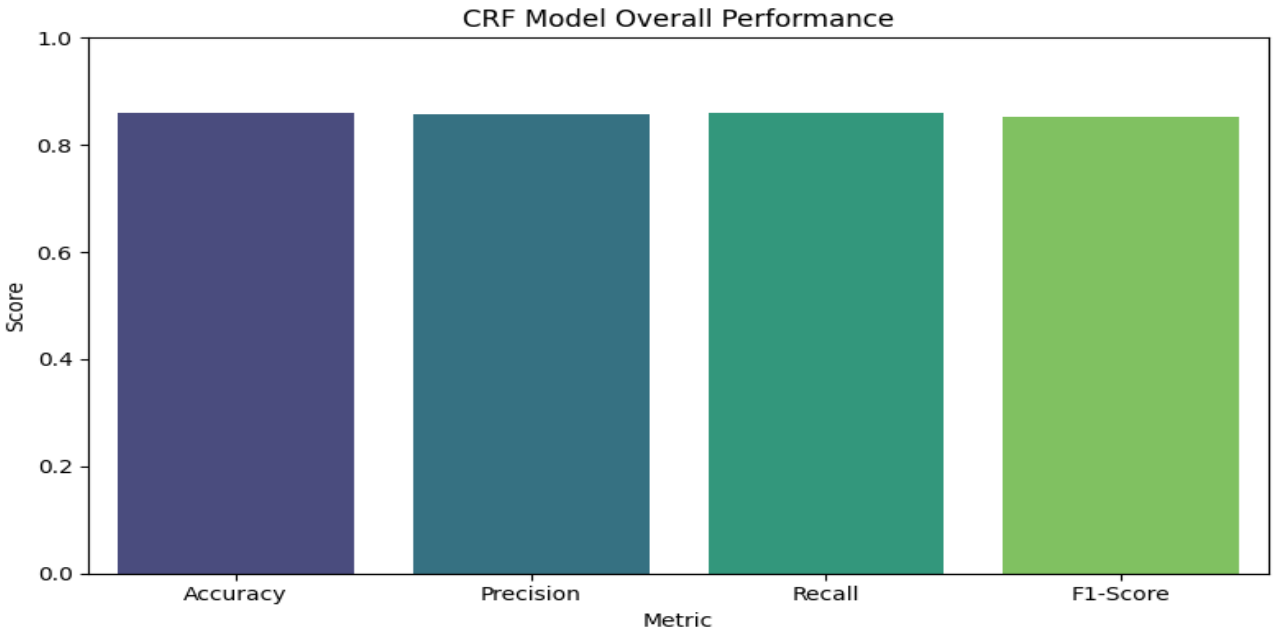


Figure 5.8 Model with CRF General Performance indicators

5.9.2 IGTRee Model Performance Metrics in General

Compared to the CRF model the IGTree model exhibits rather poor performance in Amharic text chunking. It gets a Micro F1-score of 0.518 which means it predicts more over half of the chunk labels correctly and its weighted precision is 0.696 which means the model is occasionally correct when it makes a prediction the lower recall of 0.518 shows that it misses many correct chunk labels in the data. The weighted F1-score of 0.582 shows how uneven precision and recall are.

- The IGTree model only captures Amharic chunk structures with limited recall and general moderate performance when compared to more robust models like CRF.

Table 5.8 IGTRee Model - Overall Metrics

1	Accuracy (Micro F1-score):	0.518
2	Weighted Precision:	0.696
3	Weighted Recall:	0.518
4	Weighted F1-score:	0.582

The chart indicates the IGTree Model's overall performance metrics (Accuracy, Precision, Recall and F1-Score) has been successfully generated. This chart provides a quick visual overview of how the IGTree model performs across these key metrics.

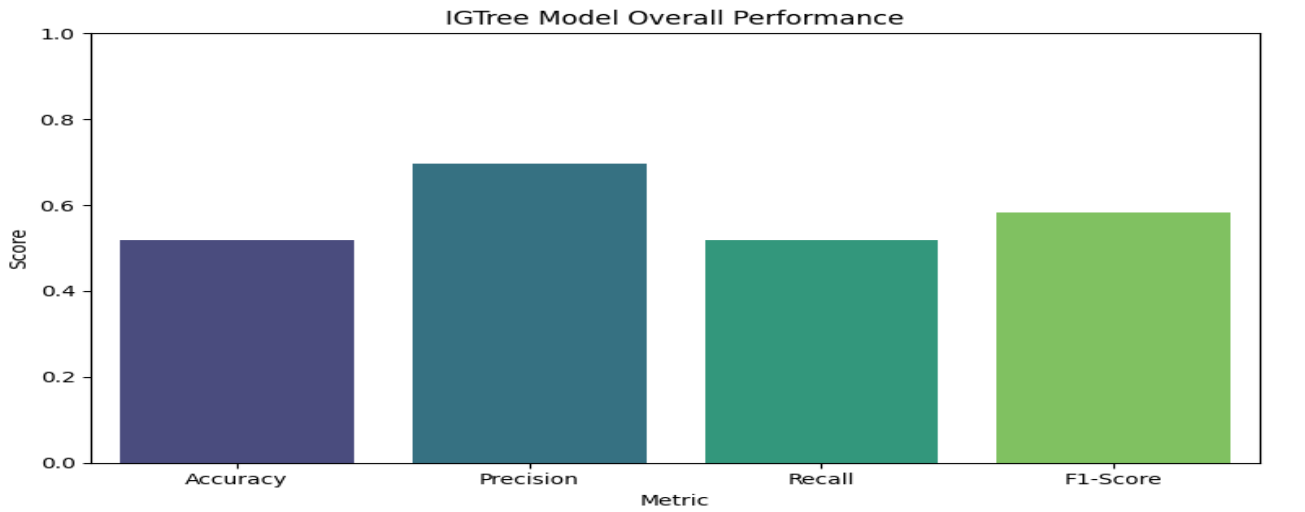


Figure 5. 9 IGTRee Model Overall Performance

5.9.3 IB1 Model General Performance Indexes

The IB1 approach exhibits mediocre Amharic text chunking performance with an accuracy (Micro F1-score) of 0.522 it accurately forecasts somewhat over half of the chunk labels its weighted precision of 0.737 shows that the model often gets it right when it makes a prediction however the weighted recall of 0.522 indicates that it overlooks a great deal of accurate chunk examples in the dataset. The weighted F1-score of 0.598 shows this trade-off between precision and recall.

- The IB1 model's prediction correctness surpasses its coverage yet its general performance is restricted in comparison to more powerful models like CRF.

Table 5.9 IB1 Model - Overall Metrics

1	Weighted Precision:	0.737
2	Weighted Recall	0.522
3	Weighted F1-score	0.598

The IB1 Model's overall performance showing its Accuracy (Micro F1-score), weighted Precision, weighted Recall and weighted F1-score as you can see the F1-score and Accuracy are around 0.52-0.6 which is lower than the CRF model's performance.

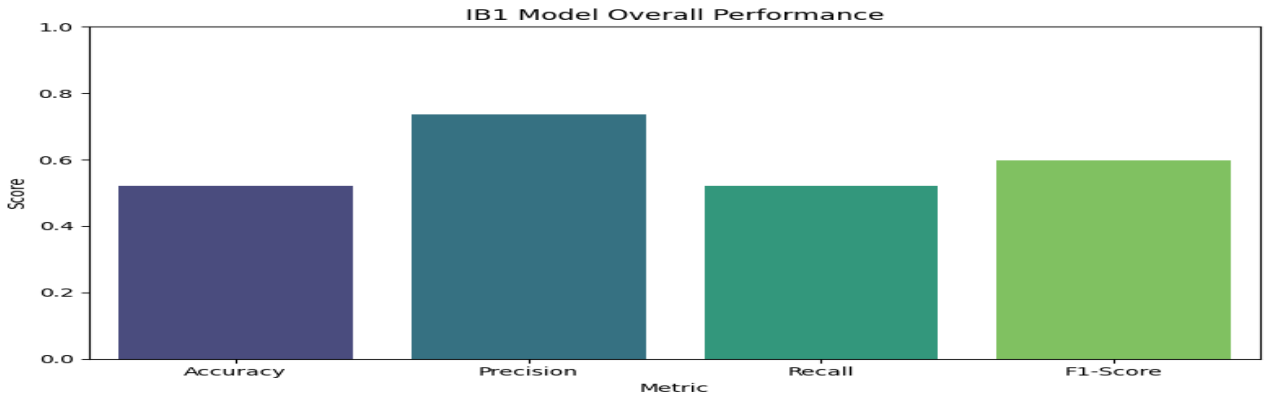


Figure 5.10 IB1 Model Overall Performance

5.9.4 SVM Model General Performance Metrics

The SVM model does rather well in Amharic text chunking it gets an accuracy (Micro F1-score) of 0.861 which means it correctly classifies most chunk labels its weighted precision is 0.854 which indicates that most of its predictions are correct. Its weighted recall of 0.861 shows that it successfully captures most of the actual chunk instances in the dataset. The weighted F1-score of 0.855 confirms a good balance between precision and recall with constant high marks throughout all metrics the SVM model shows very good overall performance hence positioning it as a powerful and aggressive Amharic text chunking option.

Table 5.10 SVM Model - Overall Metrics

1	Accuracy (Micro F1-score):	0.861
2	Weighted Precision:	0.854
3	Weighted Recall:	0.861
4	Weighted F1-score:	0.855

5.10 Test Set Chunk Tag Distribution for Evaluating the ATC Model

The chart illustrates the distribution of chunk tag occurrences in the test dataset utilized for evaluating the Amharic text chunking model. This chart displays the test data's natural and unbalanced distribution in contrast to the balanced training dataset produced by SMOTE. This is crucial because it enables the model to be evaluated in real-world language data-like conditions.

The x-axis on the chart shows 13 different IOB2 chunk labels and the y-axis shows how many times each label appears (support counts). The chart employs a bright gradient pattern that changes from dark red tones to lighter green and blue colors and the labels are somewhat tilted to improve readability. Values are also easier to read with the help of horizontal dashed grid lines.

It is obvious from the graph that the dataset is extremely unbalanced the I-NP tag is the most prevalent majority class in the test set with 1,937 instances making it the most supported tag with 1,199 occurrences the B-NP tag is also often seen. Some chunk tags contain very few samples but other classes including B-VP, I-VP and B-PP have a fair amount of support.

Extremely low frequencies are seen in minority classes like I-ADJP, I-PP and I-ADVP. Specifically in the whole test dataset there is just one instance of B-SBAR and one of I-Vp. Their bars are almost flat and hardly visible close to the baseline of the graph. It implies that the model had very little example data to evaluate these classes because the support counts are really low.

The visualization also aids in explaining certain of the outcomes observed in the ROC curve analysis. The poor ROC performance of I-Vp and B-SBAR which had an AUC of 0.50 is a good example of this and may be directly related to their very tiny support counts. The model is unable to develop reliable prediction boundaries because each class only occurs once in the test set leading to performance comparable to random guessing.

Similarly the low support count of just 12 occurrences for the I-PP ROC curve explains the observed irregular staircase pattern. A step-like curve is produced rather than a smooth one because each correct or incorrect prediction results in a significant change in the ROC computation due to the small number of samples.

The graph clearly shows that despite SMOTE's application in balancing the training data the assessment was carried out using the original unbalanced test dataset. Since it offers a more realistic assessment of how well the model performs on naturally distributed language data this is regarded as good machine learning practice. The visualization verifies the strong performance of the proposed Amharic text chunking model on key chunk classes despite challenges with very rare labels caused by restricted evaluation samples.

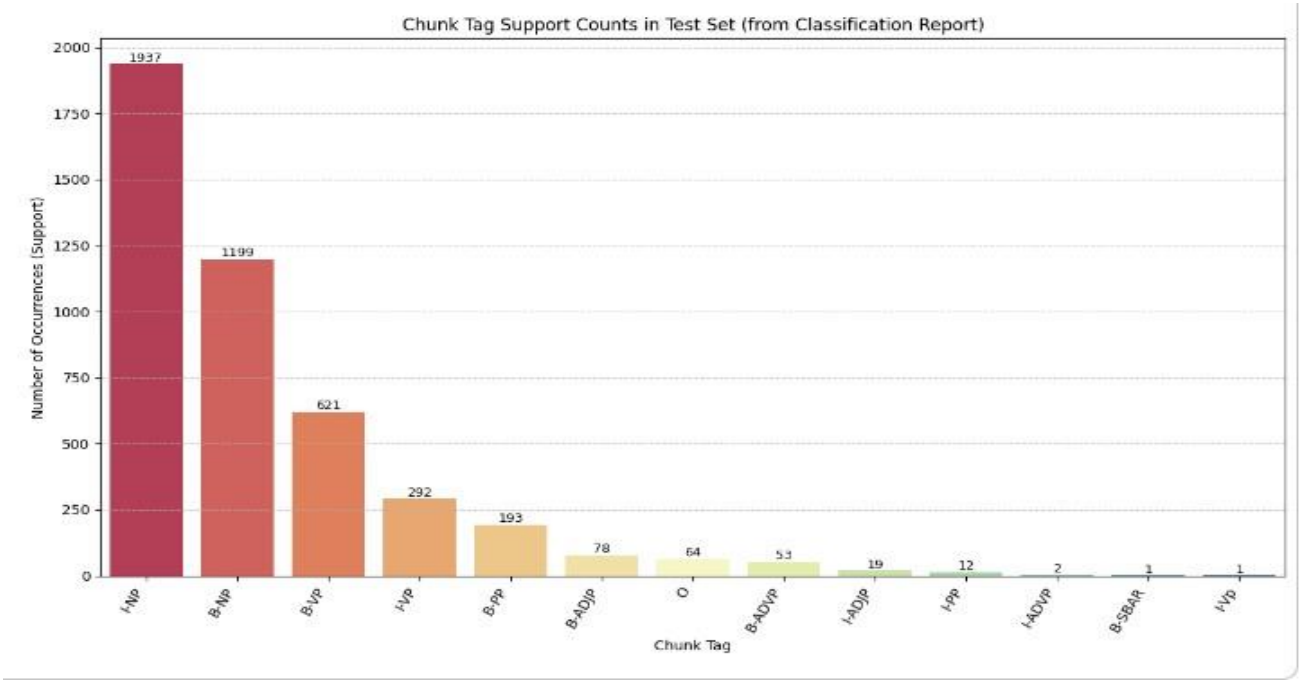


Figure 5.11 Chunk tag support counts in test set (from classification report)

5.11 Model Performance Analysis of the Study

5.11.1 General Model Performance Comparison Based on F1-score and Accuracy

The results of the experiments indicate that among Logistic Regression and Random Forest models, the suggested model CRF had the highest general performance. The suggested model's accuracy of 0.860306 and F1-score of 0.851899 are much higher than those of the other models. Random Forest gave an accuracy of 0.518043 and an F1-score of 0.581906 while Logistic Regression got an accuracy of 0.521585 and an F1-score of 0.598439.

The comparison shows that the suggested approach CRF learned chunk patterns more successfully and accurately predicted chunk labels. Its improved recall, precision, F1-score and accuracy suggest more contextual awareness and chunk prediction power for Amharic text chunking projects.

Table 5.11 Model Performance Comparison Using F1-Score and Accuracy

Model	Precision	Recall	F1-Score	Accuracy
Proposed Model	0.857931	0.860496	0.851899	0.860306
Logistic Regression	0.736655	0.521701	0.598439	0.521585
Random Forest	0.695973	0.518158	0.581906	0.518043

The chart indicates the model performance comparison (F1-Score and Accuracy) has been successfully generated and it provides a quick visual overview of model performance comparison

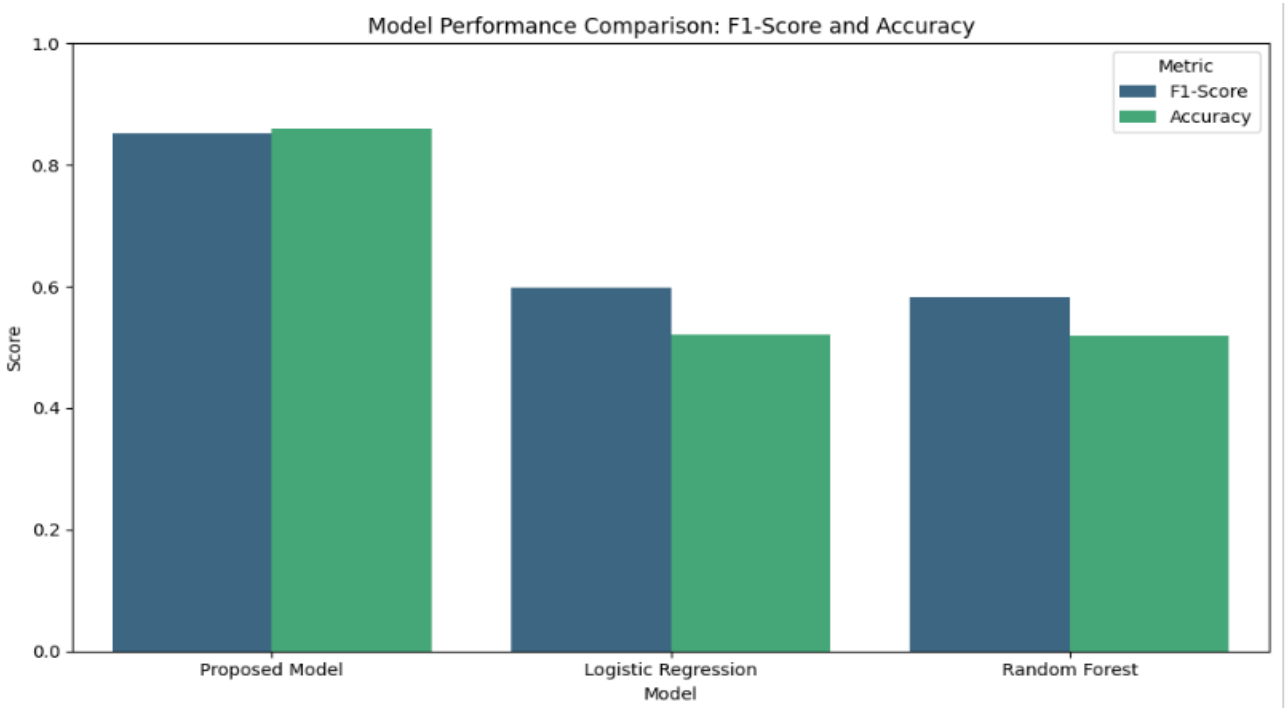


Figure 5.12 Model Performance Comparison (F1-Score and Accuracy)

5.11.2 F1-score and Accuracy Comparison of CRF, IB1 and IGTree Models

Using F1-score and accuracy for Amharic text chunking, this comparison highlights the general performance of the CRF, IB1 and IGTree models with an F1-score of 0.8519 and an accuracy of 0.8603 the CRF model clearly outperforms the other models. This suggests that in Amharic sentences CRF is far more dependable and consistent in accurately spotting chunk forms. The IB1 model with an F1-score of 0.5984 and an accuracy of 0.5216 performs reasonably but exhibits limited capacity to capture complex sentence patterns. The IGTree model with a somewhat lower F1-score of 0.5819 also demonstrates less predictive power than CRF.

- CRF is the top-performing model overall IB1 and IGTree follow behind because their accuracy is less and their sequence patterns are weaker.

Table 5.12 F1-score, Accuracy Comparison of CRF, IB1, IGTree Models

NO	Model	Metric	Value
1	CRF	F1-Score	0.851899
2	IB1	F1-Score	0.598439
3	IGTree	F1-Score	0.581906
4	CRF	Accuracy	0.860306
5	IB1	Accuracy	0.521585

Here is a bar chart visualizing the F1-Score and Accuracy for each model (CRF, IB1, and IGTree).



Figure 5.13 F1-score, Accuracy Comparison of CRF, IB1 and IGTree Models

5.11.3 Performance of F1-Scores per Label across Models Compared

This comparison illustrates how three models CRF, IB1 and IGTree do on various Amharic text chunk.

- The CRF model generally runs the best over almost all chunk types particularly on the most crucial and common ones. For B-VP (0.942), I-NP (0.884) and I-VP (0.889) it produces very strong results so demonstrating its great effectiveness in identifying both noun and verb phrase constructions in Amharic sentences.
- In contrast both IB1 and IGTree models do not perform very well on common chunks like NP and PP their scores are always lower than CRF's for instance IB1 and IGTree find verb phrase detection and general sequence consistency to be more difficult.
- Less common or sophisticated chunk kinds of all versions exhibit lower performance labels including ADJP, I-PP and I-ADJP have low ratings across models while B-SBAR and I-ADVP are not learned at all (score is 0) therefore pointing to a lack of adequate training data for these patterns.
- The "O" (outside chunk) label also shows a sharp difference CRF performs reasonably well (0.750) whereas IB1 and IGTree fare quite poorly therefore indicating they battle to accurately identify non-chunk words for Amharic chunking particularly for basic sentence patterns CRF is the most trustworthy model IB1 and IGTree fall behind mostly because of limited data and weaker sequence learning capabilities.

Table 5.13 Detailed F1-Scores per Label across Models

Chunk Type	CRF F1-Score	IB1 F1-Score	IGTree F1-Score
B-NP	0.824	0.539	0.529
I-NP	0.884	0.689	0.694
B-VP	0.942	0.540	0.536
B-PP	0.822	0.747	0.648
I-VP	0.889	0.627	0.546
B-ADVP	0.772	0.696	0.520
O	0.750	0.061	0.059
B-ADJP	0.233	0.254	0.129
I-PP	0.182	0.000	0.000
I-ADJP	0.182	0.000	0.000
B-SBAR	0.000	0.000	0.000
I-ADVP	0.000	0.000	0.000

5.11.4 Model Precision, Recall, F1-score Accuracy

IB1 and IGTree both exhibit comparable reduced performance levels against CRF. This graphic shows how well the CRF uses sequence dependencies and contextual information for more exact chunking.

Table 5.14 Model Precision, Recall, F1-score Accuracy.

No	Model	Precision	Recall	F1-Score	Accuracy
1	CRF	0.857931	0.860496	0.851899	0.860306
2	IB1	0.736655	0.521701	0.598439	0.521585
3	IGTree	0.695973	0.518158	0.581906	0.518043

The diagram below show Performance Comparison of CRF, IB1 and IGTree including Model, Precision,

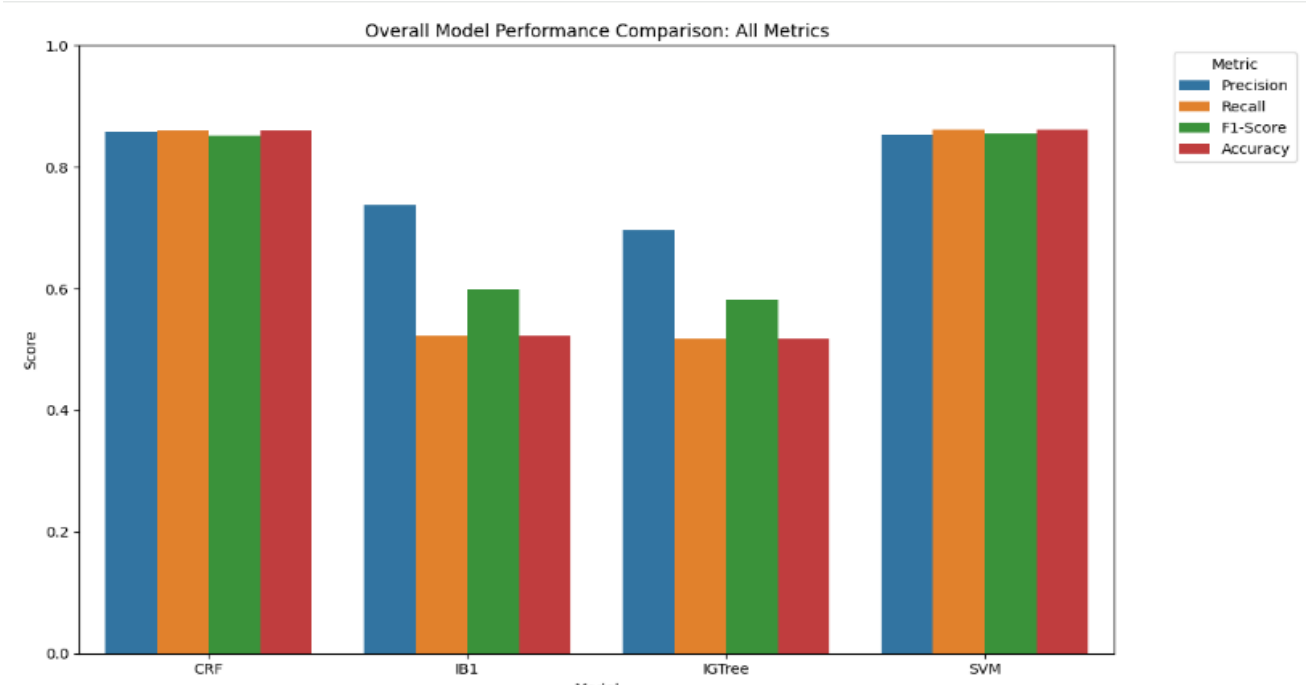


Figure 5.14 Model Precision, Recall, F1-score Accuracy

5.11.5 Comparing General Model Performance (CRF, IB1, IGTree, SVM)

The models have effectively contrasted general performance using Accuracy and F1-Score

- SVM and CRF are the best performers the Support Vector Machine (SVM) model had the highest F1-Score of 0.855 and Accuracy of 0.861 closely followed by the Conditional Random Field (CRF) model with an F1-Score of 0.852 and Accuracy of 0.860.
- Significant gap with simpler models with F1-Scores about 0.60 both IB1 and IGTree models show much lower performance on all metrics hence demonstrating that for this challenging chunking job simpler instance-based approaches are less effective than sequence-aware models like CRF or strong classifiers like SVM when paired with rich features.

Table 5.15 Comparing General Model Performance (CRF, IB1, IGTree, SVM)

Model	Precision	Recall	F1-Score	Accuracy
CRF	0.857931	0.860496	0.851899	0.860306
IB1	0.736655	0.521701	0.598439	0.521585
IGTree	0.695973	0.518158	0.581906	0.518043
SVM	0.853850	0.860939	0.854586	0.860748

The diagram below show Overall Model Performance Comparison (CRF, IB1, IGTree, and SVM)

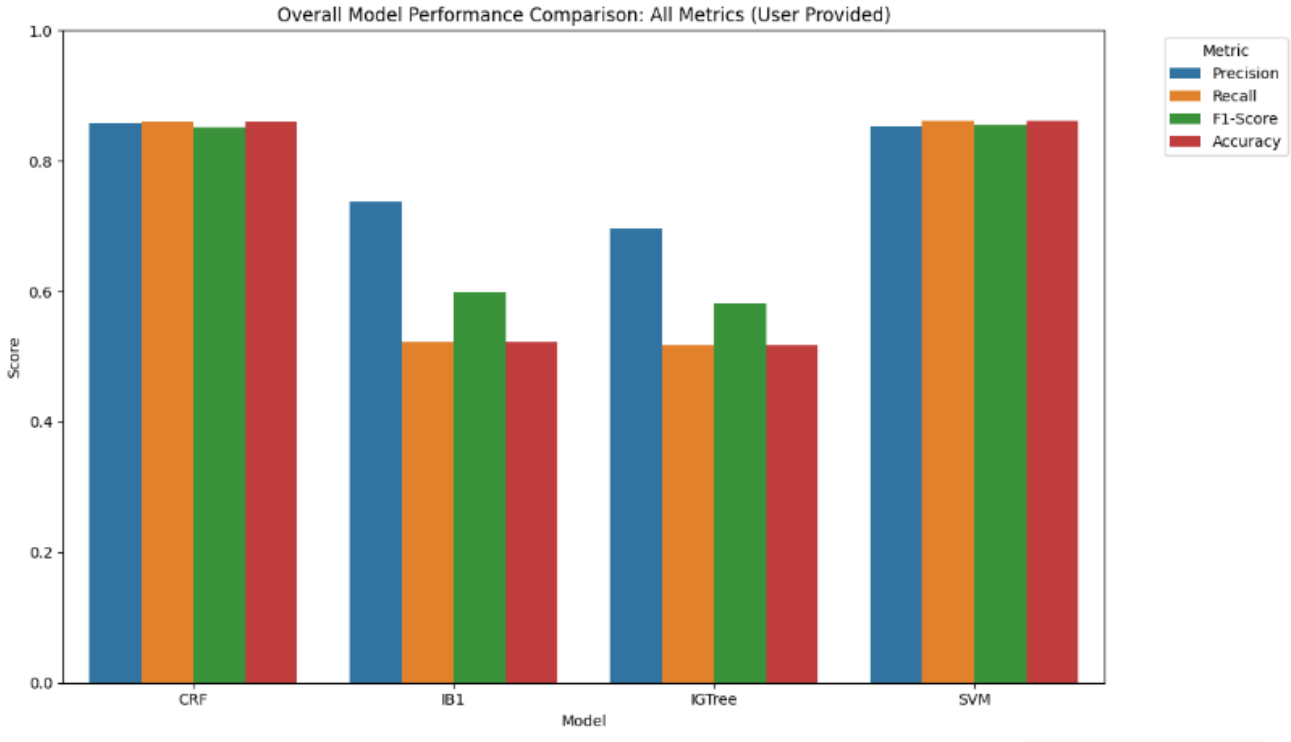


Figure 5.15 Comparing General Model Performance (CRF, IB1, IGTree, SVM)

CHAPTER SIX

6. CONCLUSION AND RECOMMENDATION

6.1 Conclusion

This study successfully investigated the development of an Amharic text chunker using machine learning techniques. The research addressed important limitations found in previous Amharic chunking studies, particularly the limited use of modern machine learning methods and the shortage of annotated datasets.

The study designed and implemented chunking models using Conditional Random Fields (CRF) and Memory-Based Learning (MBL) approaches. Through a series of experiments the research demonstrated that supervised machine learning models are effective in identifying meaningful phrase structures such as noun phrases, verb phrases, adjective phrases and prepositional phrases in Amharic sentences.

The results showed that the CRF-based model performed better than the MBL models because it effectively captures sequential relationships and contextual dependencies between words. In addition, the integration of Part-of-Speech tags and contextual features significantly improved chunking accuracy and model performance.

Another important achievement of this research is the preparation of a manually annotated Amharic chunking dataset using the IOB2 tagging format. This dataset can serve as a valuable resource for future studies in Amharic Natural Language Processing.

Generally, the study demonstrated that machine learning-based chunking systems can provide a strong foundation for developing advanced Amharic language technologies such as machine translation, information extraction and question answering systems, text summarization, and information retrieval applications. Therefore, this research contributes to the advancement of computational linguistic studies for Amharic and other Ethiopian languages.

6.2 Recommendation

Based on the findings obtained from this study several recommendations are suggested for future research and development activities.

1. Future researchers should prepare larger and more balanced annotated datasets that include different types of Amharic text from both formal and informal domains. A larger dataset can improve the learning ability and generalization performance of machine learning models.
2. It is also recommended to investigate additional linguistic features such as morphological information, semantic representations and word embeddings to further improve chunking accuracy

Since Amharic is morphologically rich incorporating morphological analysis may help the system better understand complex word structure

3. Researchers should also explore modern deep learning approaches such as BiLSTM, BiLSTM-CRF and Transformer-based architectures for Amharic text chunking. These approaches have shown strong performance in many Natural Language Processing tasks and may produce better results for Amharic as well.
4. In addition universities, research institutions and government organizations should support the development of publicly available Amharic NLP resources including corpora annotation tools and language processing frameworks. This support can help accelerate research activities and encourage the development of language technologies for Ethiopian languages.

6.3 Future Work

Although this study achieved promising results there are still several areas that can be improved in future research.

1. Future studies can focus on expanding the size and diversity of the dataset by including more text sources such as social media, online discussions, spoken language transcripts and conversational texts. This can help improve the robustness of the chunking system when handling informal language.
2. The current study mainly focused on CRF and MBL approaches therefore future research can investigate advanced deep learning methods such as BiLSTM, GRU, attention mechanisms and Transformer-based models for improving chunking performance.
3. Future work can also integrate morphological analyzers and semantic feature extraction techniques to better handle the complex grammatical structure of Amharic. Since Amharic is highly inflected and morphologically rich these techniques may improve the system's understanding of sentence structures.
4. Additionally, the developed chunker can be extended into a full syntactic parser capable of performing deeper sentence analysis. Future researchers may also evaluate the chunker within practical applications such as machine translation, information retrieval, speech processing, intelligent tutoring systems and question answering systems.
5. Finally, similar machine learning approaches can be adapted and extended to other Ethiopian languages that currently lack sufficient Natural Language Processing resources and tools.

REFERENCE

4715182, T. B.-A. at S. (n.d.). Phonotactic Constraints between Amharic and English: Implications for Language Learning. *Papers.Ssrn.Com*. Retrieved June 9, 2025, from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4715182

Abeba, I. (2013). *A Hybrid Approach to Amharic Base Phrase Chunking and Parsing*. http://thesisbank.jhia.ac.ke/4220/

Abolohom, A., Omar, N., Pais, S., & Cordeiro, J. (2021). A Comparative Study of Linguistic and Computational Features Based on a Machine Learning for Arabic Anaphora Resolution. *Procedia CIRP*, *189*(2019), 37–47. https://doi.org/10.1016/j.procs.2021.05.068

AlBadani, B., Shi, R., & Dong, J. (2022a). A Novel Machine Learning Approach for Sentiment Analysis on Twitter Incorporating the Universal Language Model Fine-Tuning and SVM. *Applied System Innovation*, *5*(1). https://doi.org/10.3390/asi5010013

AlBadani, B., Shi, R., & Dong, J. (2022b). A Novel Machine Learning Approach for Sentiment Analysis on Twitter Incorporating the Universal Language Model Fine-Tuning and SVM. *Applied System Innovation*, *5*(1). https://doi.org/10.3390/asi5010013

Ali, W., Malik, M., Hussain, S., Shahid, S., & Ali, A. (2010). *Urdu noun phrase chunking*. https://www.researchgate.net/profile/Wajid-Ali-21/publication/202401984_Urdu_Noun_Phrase_chunking_HMM_based_approach/links/5874cc0208aebf17d3b3b09f/Urdu-Noun-Phrase-chunking-HMM-based-approach.pdf

Alnagi, E., Ghnemat, R., & Abu Al-Haija, Q. (2025a). Boosting Arabic text classification using hybrid deep learning approach. *Discover Applied Sciences*, *7*(6). https://doi.org/10.1007/s42452-025-07025-x

Alnagi, E., Ghnemat, R., & Abu Al-Haija, Q. (2025b). Boosting Arabic text classification using hybrid deep learning approach. *Discover Applied Sciences*, *7*(6). https://doi.org/10.1007/s42452-025-07025-x

Balek, V., Sýkora, L., Sklenák, V., & Kliegr, T. (2025a). LLM-based feature generation from text for interpretable machine learning. *Machine Learning*, *114*(11), 1–35. https://doi.org/10.1007/s10994-025-06867-1

Balek, V., Sýkora, L., Sklenák, V., & Kliegr, T. (2025b). *LLM-based feature generation from text for interpretable machine learning*. https://doi.org/10.1007/s10994-025-06867-1

Chen, W., Zhang, Y., NLP, H. I.-, & 2006, undefined. (n.d.-a). Chinese Chunking based on Conditional Random Fields. *Anlp.Jp*.

Chen, W., Zhang, Y., NLP, H. I.-, & 2006, undefined. (n.d.-b). Chinese Chunking based on Conditional Random Fields. *Anlp.Jp*. Retrieved June 10, 2025, from https://www.anlp.jp/proceedings/annual_meeting/2006/pdf_dir/E1-8.pdf

Dahia, I., & Belbacha, M. (2024). *International Journal of Linguistics, Literature and Translation Machine-Learning-based English Quranic Translation: An Evaluation of ChatGPT*. https://doi.org/10.32996/ijllt

El Mir, I., El Kafhali, S., & Haqiq, A. (2022). *A Hybrid Learning Approach for Text Classification Using Natural Language Processing* (pp. 428–439). https://doi.org/10.1007/978-3-031-07969-6_32

Fake, A., Detection, A., & Textual, U. (2024). *MSc. Thesis by Tigist Shiferaw Shitie Program : Information Technology Advisor : Zewdie Mossie (Ph. D) Tigist Shiferaw Shitie Advisor : Zewdie Mossie (Ph. D)*.

Gunasekaran, K. P. (n.d.). *Exploring Sentiment Analysis Techniques in Natural Language Processing: A Comprehensive Review*.

Gupta, V., & Gupta, E. (2022). Punjabi news multi-classification using language generation-based optimized long short-term memory networks. *Evolving Systems*, *14*, 1–10. https://doi.org/10.1007/s12530-022-09428-2

Hossain, M. (2022). *Support Vector Machine**.

Hwang, Y., Chung, H., Park, S. … Y. K.-J. of K., & 2002, undefined. (n.d.). Improving the performance of Korean text chunking by machine learning approaches based on feature set selection. *Koreascience.Kr*. Retrieved June 10, 2025, from https://koreascience.kr/article/JAKO200211921392561.page

Ibrahim, A., & Assabie, Y. (2016). Hierarchical Amharic base phrase chunking using HMM with error pruning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9561 LNCS*, 126–135. https://doi.org/10.1007/978-3-319-43808-5_10

Kanwal, I., Wahid, F., Ali, S., Rehman, A. U., Alkhayyat, A., & Al-Radaei, A. (2023). Sentiment Analysis Using Hybrid Model of Stacked Auto-Encoder-Based Feature Extraction and Long Short-Term Memory-Based Classification Approach. *IEEE Access*, *11*, 124181–124197. https://doi.org/10.1109/ACCESS.2023.3313189

Khoufi, N., Aloulou, C., 11th, L. B.-2014 I., & 2014, undefined. (n.d.). Chunking Arabic texts using conditional random fields. *Ieeexplore.Ieee.OrgN Khoufi, C Aloulou, LH Belguith2014 IEEE/ACS 11th International Conference on Computer Systems, 2014•ieeexplore.Ieee.Org*.

Mishra, S., Aggarwal, M., Yadav, S., & Sharma, Y. (2023). An Automated Model for Sentimental AnalysisUsing Long Short-Term Memory-based DeepLearning Model. *International Journal of Engineering and Manufacturing*, *13*(5), 11–20. https://doi.org/10.5815/ijem.2023.05.02

Models, M., & Haugh, M. (n.d.). *Machine Learning for OR & FE*.

Mogridge, R., & Ragni, A. (2024). Learning from memory-based models. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2360–2364. https://doi.org/10.21437/Interspeech.2024-1628

Mohammed Tigabu, K. (2023). *AMHARIC ANAPHORA RESOLUTION USING MACHINE LEARNING APPROACH: THE CASE OF INDEPENDENT AND HIDDEN ANAPHORA RESOLUTION MSc THESIS*. (March).

Molina, A., Research, F. P.-J. of M. L., & 2002, U. (n.d.). Shallow parsing using specialized HMMsMolina, A., Research, F. P.-J. of M. L., & 2002, undefined. (n.d.). Shallow parsing using specialized HMMs. Jmlr.OrgA Molina, F PlaJournal of Machine Learning Research, 2002•jmlr.Org. Retrieved June 7, 2025, from http. *Jmlr.OrgA Molina, F PlaJournal of Machine Learning Research, 2002•jmlr.Org*. Retrieved June 7, 2025, from https://www.jmlr.org/papers/v2/molina02a.html

Mulu, A., & Goyal, V. (2020). Amharic Phrase Chunking with Conditional Random Fields. *Journal of Scientific Research*, *64*(01), 153–159. https://doi.org/10.37398/jsr.2020.640135

Qi, Y., & Shabrina, Z. (2023). Sentiment analysis using Twitter data: a comparative application of lexicon- and machine-learning-based approach. *Social Network Analysis and Mining*, *13*(1). https://doi.org/10.1007/s13278-023-01030-x

Rekha Raj, C. T., & Reghu Raj, P. C. (2016). Text chunker for Malayalam using Memory-Based Learning. *2015 International Conference on Control, Communication and Computing India, ICCC 2015*, 595–599. https://doi.org/10.1109/ICCC.2015.7432966

Report, I. L. K. T. (2018). *TiMBL : Tilburg Memory-Based Learner*.

Sarkar, K., on, V. G.-2014 2nd I. C., & 2014, undefined. (n.d.). Bengali noun phrase chunking based on conditional random fields. *Ieeexplore.Ieee.OrgK Sarkar, V Gayen2014 2nd International Conference on Business and Information, 2014•ieeexplore.Ieee.Org*. Retrieved June 7, 2025, from https://ieeexplore.ieee.org/abstract/document/6970957/

Sharma, S. … G. L.-C. on C. S. and, & 2011, undefined. (n.d.). Using Hidden Markov Model to improve the accuracy of Punjabi POS tagger. *Ieeexplore.Ieee.Org*.

Shiferaw, T., Advisor, S., Mossie, Z., & Markos, D. (2024). *AMHARIC FAKE ACCOUNT DETECTION USING TEXTUAL INFORMATION:-DEEP LEARNING APPROACH*.

Sietsma, A. J., Ford, J. D., & Minx, J. C. (2024). The next generation of machine learning for tracking adaptation texts. *Nature Climate Change*, *14*(1), 31–39. https://doi.org/10.1038/s41558-023-01890-3

Stammbach, D., & Ash, E. (2022). DocSCAN: Unsupervised Text Classification via Learning from Neighbors. *KONVENS 2022 - Proceedings of the 18th Conference on Natural Language Processing*, (Konvens), 21–28.

Suraksha, N. … K. R.-… C. on intelligent, & 2017, undefined. (n.d.). Part-of-speech tagging and parsing of Kannada text using Conditional Random Fields (CRFs). *Ieeexplore.Ieee.OrgNM Suraksha, K Reshma, KMS Kumar2017 International Conference on Intelligent Computing and Control, 2017•ieeexplore.Ieee.Org*. Retrieved June 7, 2025, from https://ieeexplore.ieee.org/abstract/document/8321833/

TIGABU, K. (2023). *AMHARIC ANAPHORA RESOLUTION USING MACHINE LEARNING APPROACH: THE CASE OF INDEPENDENT AND HIDDEN ANAPHORA RESOLUTION*. http://ir.haramaya.edu.et/hru/bitstream/handle/123456789/6722/Msc%20Thesis%20%20Kedir%20M.pdf?sequence=1

Upgris, G. B.-J. I., & 2021, undefined. (n.d.). AMHARIC TEXT CHUNKER USING CONDITIONAL RANDOM FIELDS. *Journal.Upgris.Ac.Id*. Retrieved June 7, 2025, from http://journal.upgris.ac.id/index.php/JIU/article/view/9086

Wang, J., Wei, J., Tian, F., & Wei, Y. (2024). A comparative study of machine learning models for sentiment analysis of transboundary rivers news media articles. *Soft Computing*, *28*(23), 13331–13347. https://doi.org/10.1007/s00500-024-10357-2

Wang, Z., Liu, M., & Liu, K. (2024). Utilizing Machine Learning Techniques for Classifying Translated and Non-Translated Corporate Annual Reports. *Applied Artificial Intelligence*, *38*(1). https://doi.org/10.1080/08839514.2024.2340393

Xu, J., Li, J., Liu, Z., Suryanarayanan, N. A. V., Zhou, G., Guo, J., Iba, H., & Tei, K. (2024). Large Language Models Synergize with Automated Machine Learning. *Transactions on Machine Learning Research*, *2024*, 1–49.

Yimam, S. M., Ayele, A. A., Venkatesh, G., Gashaw, I., & Biemann, C. (2021). Introducing various semantic models for amharic: Experimentation and evaluation with multiple tasks and datasets. *Future Internet*, *13*(11). https://doi.org/10.3390/fi13110275

Zain, M., Hussain, N., Qasim, A., Mehak, G., Ahmad, F., Sidorov, G., & Gelbukh, A. (2025a). RU-OLD: A Comprehensive Analysis of Offensive Language Detection in Roman Urdu Using Hybrid Machine Learning, Deep Learning, and Transformer Models. *Algorithms*, *18*(7). https://doi.org/10.3390/a18070396

Zain, M., Hussain, N., Qasim, A., Mehak, G., Ahmad, F., Sidorov, G., & Gelbukh, A. (2025b). RU-OLD: A Comprehensive Analysis of Offensive Language Detection in Roman Urdu Using Hybrid Machine Learning, Deep Learning, and Transformer Models. *Algorithms*, *18*(7). https://doi.org/10.3390/a18070396

Zhao, Y., Fuzzy, T. Z.-2015 12th I. C. on, & 2015, undefined. (n.d.). Exploiting clause boundary information as features for Chinese functional chunk parsing. *Ieeexplore.Ieee.OrgY Zhao, T Zhao2015 12th International Conference on Fuzzy Systems and Knowledge, 2015•ieeexplore.Ieee.Org*. Retrieved June 7, 2025, from https://ieeexplore.ieee.org/abstract/document/7382058/

Zhao, Y. R., & Chen, D. (2024). HMM Quantifying investor sentiment: Research on sentiment factors based on text analysis and Hidden Markov Model. In *ACM International Conference Proceeding Series* (Vol. 1, Number 1). Association for Computing Machinery. https://doi.org/10.1145/3675417.3675522

Zhou, Q. (n.d.). *Chapter 3 Hidden Markov Models (HMMs)*. 33–41.

List of Appendix

Appendix 1: - Amharic ፊደል (Fidel or alphabet) Unicode standard

Table 1.1. Amharic ፊደል (Fidel or alphabet) Unicode standard

	120	121	122	123	124	125	126	127	128	129	12A	12B
0	ሀ	ሐ	ሠ	ሰ	ቀ	ቐ	በ	ተ	ኀ	ነ	አ	ኰ
1	ሁ	ሑ	ሡ	ሱ	ቁ	ቑ	ቡ	ቱ	ኁ	ኑ	ኡ	
2	ሂ	ሒ	ሢ	ሲ	ቂ	ቒ	ቢ	ቲ	ኂ	ኒ	ኢ	ኩ
3	ሃ	ሓ	ሣ	ሳ	ቃ	ቓ	ባ	ታ	ኃ	ና	ኣ	ኪ
4	ሄ	ሔ	ሤ	ሴ	ቄ	ቔ	ቤ	ቴ	ኄ	ኔ	ኤ	ኬ
5	ህ	ሕ	ሥ	ስ	ቅ	ቕ	ብ	ት	ኅ	ን	እ	ክ
6	ሆ	ሖ	ሦ	ሶ	ቆ	ቖ	ቦ	ቶ	ኆ	ኖ	ኦ	
7	ሇ	ሗ	ሧ	ሷ	ቇ		ቧ	ቷ	ኇ	ኗ	ኧ	
8	ለ	መ	ረ	ሸ	ቈ	ቘ	ቨ	ቸ	ኈ	ኘ	ከ	ኸ
9	ሉ	ሙ	ሩ	ሹ			ቩ	ቹ		ኙ	ኩ	ኹ
A	ሊ	ሚ	ሪ	ሺ	ቊ	ቚ	ቪ	ቺ	ኊ	ኚ	ኪ	ኺ
B	ላ	ማ	ራ	ሻ	ቋ	ቛ	ቫ	ቻ	ኋ	ኛ	ካ	ኻ
C	ሌ	ሜ	ሬ	ሼ	ቌ	ቜ	ቬ	ቼ	ኌ	ኜ	ኬ	ኼ
D	ል	ም	ር	ሽ	ቍ	ቝ	ቭ	ች	ኍ	ኝ	ክ	ኽ
E	ሎ	ሞ	ሮ	ሾ			ቮ	ቾ		ኞ	ኮ	ኾ
F	ሏ	ሟ	ሯ	ሿ			ቯ	ቿ		ኟ	ኯ	
	12c	12d	12e	12f	130	131	132	133	134	135	136	137
0	ዀ	ዐ	ዠ	ደ	ጀ	ጐ	ጠ	ጰ	ፀ	ፐ	※	፩
1A		ዑ	ዡ	ዱ	ጁ		ጡ	ጱ	ፁ	ፑ	፡	፪
2	ዂ	ዒ	ዢ	ዲ	ጂ	ጒ	ጢ	ጲ	ፂ	ፒ	።	፫
3	ዃ	ዓ	ዣ	ዳ	ጃ	ጓ	ጣ	ጳ	ፃ	ፓ	፣	፬
4	ዄ	ዔ	ዤ	ዴ	ጄ	ጔ	ጤ	ጴ	ፄ	ፔ	፤	፭
5	ዅ	ዕ	ዥ	ድ	ጅ	ጕ	ጥ	ጵ	ፅ	ፕ	፥	፮
6		ዖ	ዦ	ዶ	ጆ		ጦ	ጶ	ፆ	ፖ	፦	፯
7			ዧ	ዷ	ጇ		ጧ	ጷ	ፇ	ፗ	፧	፰
8	ወ	ዘ	የ	ጀ	ገ	ኘ	ጨ	ጸ	ፈ	ፘ	፨	፱
9	ዉ	ዙ	ዩ	ጁ	ጉ	ኙ	ጩ	ጹ	ፉ	ፙ	፩	፲
A	ዊ	ዚ	ዪ	ጂ	ጊ	ኚ	ጪ	ጺ	ፊ	ፚ	፪	፳
B	ዋ	ዛ	ያ	ጃ	ጋ	ኛ	ጫ	ጻ	ፋ		፫	፴
C	ዌ	ዜ	ዬ	ጄ	ጌ	ኜ	ጬ	ጼ	ፌ		፬	
D	ው	ዝ	ይ	ድ	ግ	ኝ	ጭ	ጽ	ፍ		፭	
E	ዎ	ዞ	ዮ	ጆ	ጎ	ኞ	ጮ	ጾ	ፎ		፮	
F	ዏ	ዟ	ዯ	ጇ	ጏ	ኟ	ጯ	ጿ	ፏ	፟		

Appendix 2: - POS tags used by this study

Table 2.2. POS tags used by this study

No	POS Tag	Description	Example from dataset
1.	Noun phrase	Group of words acting as a noun	በሚዛን, የምክክር
2.	Noun	Name of person, place, or thing	አማን, ከተማ
3.	NUMCR	Cardinal number (quantity)	በ2017, 2017
4.	Conjunction	Connects words or sentences	እና, እንዲሁም
5.	Numeral	Represents a number	በ2018, 25/2018
6.	Verb	Shows action or state	ተካሄደ, የተገኙ
7.	Preposition	Shows relation before a noun	እንደ, በህዝቡ
8.	Adjective	Describes a noun	ለብሔራዊ, ወቅታዊ, መሰረታዊ
9.	Postposition	Comes after noun to show relation	ጋር, ላይ
10.	Determiner	Introduces or specifies a noun	ሁሉም, እያንዳንዱ
11.	Adverb	Modifies verb, adjective, or adverb	አጠናክሮ, በቀጣይም
12.	Date	Represents date or time expression	27/2018, ዓ/ም
13.	Pronoun	Replaces a noun	እኛ, ይህንን
14.	Number	Numeric value or expression	14
15.	Preposition Phrase		
16.	Title	Indicates rank or position	አቶ
17.	Verb phrase	Group of words acting as a verb	በሚቀይር
18.	Proper Noun	Specific name of person/place	ዐቢይ, አሕመድ
19.	Foreign	Word from another language	Coffee
20.	Abbreviation	Shortened form of a word	(ቤ.ሽ.ቱ.)
21.	Hashtag	Word with # symbol	#ኢዜአ
22.	Num	Short form for number	ሚሊዮን
23.	Adposition	Preposition or postposition	በላይ
24.	Adjective		
25.	Numeral/Noun	Acts as numeral or noun	16/2018ዓ/ም
26.	Noun/Punctuation	Acts as noun or punctuation	(ቤሽቱ)
27.	Hashtag/Adj	Hashtag used as adjective	#መልካም
28.	Punctuation	Symbols like. ,!,?	፣
29.	Adj (Adjective)	describes a noun	(e.g., *big, red*)
30.	Adv (Adverb)	describes a verb/adjective	(e.g., *quickly*)
31.	Conj (Conjunction)	connects words or phrases	(e.g., *and, but*)
32.	Preposition + Noun	preposition followed by nouns	(e.g., *in house*)
33.	Postp (Postposition):	comes after noun	(common in some languages like Amharic)
34.	Prepositional Phrase	preposition + noun phrase	(e.g., *in the room*)

Appendix 3: - Chunk Tag used by this study

Table 3.3. Chunk Tag used by this study

No	Chunk tag	Description	Example from dataset
1	B-NP	Beginning of noun phrase	ሚዛን *(in: ሚዛን አማን ከተማ)*
2	I-NP	Inside noun phrase	አማን *(in: ሚዛን አማን ከተማ)*
3	B-VP	Beginning of verb phrase	ተካሂደ *(in: መድረክ ተካሂደ)*
4	O	Outside any phrase	እና
5	I-VP	Inside verb phrase	እየሰራ *(in: እየሰራ ነው)*
6	B-PP	Beginning of preposition phrase	በ *(in: በ2017 ዓ/ም)*
7	B-ADJP	Beginning of adjective phrase	በተሻለ *(in: በተሻለ ፍጥነት)*
8	B-ADVP	Beginning of adverb phrase	በጥሩ *(in: በጥሩ የሀሳብ መድረክ)*
9	I-ADJP	Inside adjective phrase	ፍጥነት *(in: በተሻለ ፍጥነት)*
10	I-PP	Inside preposition phrase	2017 *(in: በ2017 ዓ/ም)*
11	I-ADVP	Inside of adverb phrase	
12	B-SBAR	Beginning of subordinate clause	እንደ *(in: እንደ ሀገር ለብሔራዊ ምክክር)*
13	I-Vp		

Appendix 4: - POS-tagged and chunk annotated document for the training data set

Amharic Sentence

በሚዛን አማን ከተማ አስተዳደር በ2017 ዓ/ም እና በ2018 ዓ/ም የምክክር መድረክ ተካሄደ በዉይይቱም እንደ ሀገር ለብሔራዊ ምክክር ኮምሽን እንደ ግብዓት የሚያስፈልጉ ሃሳቦች የተገኙ ሲሆን መድረኩም በጥሩ የሀሳብ የምክክር መድረክ ልምድ የተወሰደበት ነው።

Sample chunked text Output

Word	POS Tags	Chunk Tags
በሚዛን	Noun Phrase	B-NP
አማን	Noun	I-NP
ከተማ	Noun	I-NP
አስተዳደር	Noun	I-NP
በ2017	NUMCR	B-NP
ዓ/ም	Noun	I-NP
እና	Conjunction	O
በ2018	NUMCR	B-NP
ዓ/ም	Noun	I-NP
የምክክር	Noun Phrase	B-NP
መድረክ	Noun	I-NP
ተካሄደ	Verb	B-VP
በዉይይቱም	Noun Phrase	B-NP
እንደ	Preposition	O
ሀገር	Noun	B-NP
ለብሔራዊ	Adjective	I-NP
ምክክር	Noun	I-NP
ኮምሽን	Noun	I-NP
እንደ	Preposition	O
ግብዓት	Noun	B-NP
የሚያስፈልጉ	Verb	B-VP
ሃሳቦች	Noun	B-NP
የተገኙ	Verb	I-VP
ሲሆን	Conjunction	O
መድረኩም	Noun	B-NP
በጥሩ	Adjective	B-NP
የሀሳብ	Noun Phrase	I-NP
የምክክር	Noun Phrase	I-NP
መድረክ	Noun	I-NP
ልምድ	Noun	I-NP
የተወሰደበት	Verb	B-VP
ነው	AUX	I-VP
።	Punctuation	O

Appendix 5: - Sample chunk tagged document for the training data set

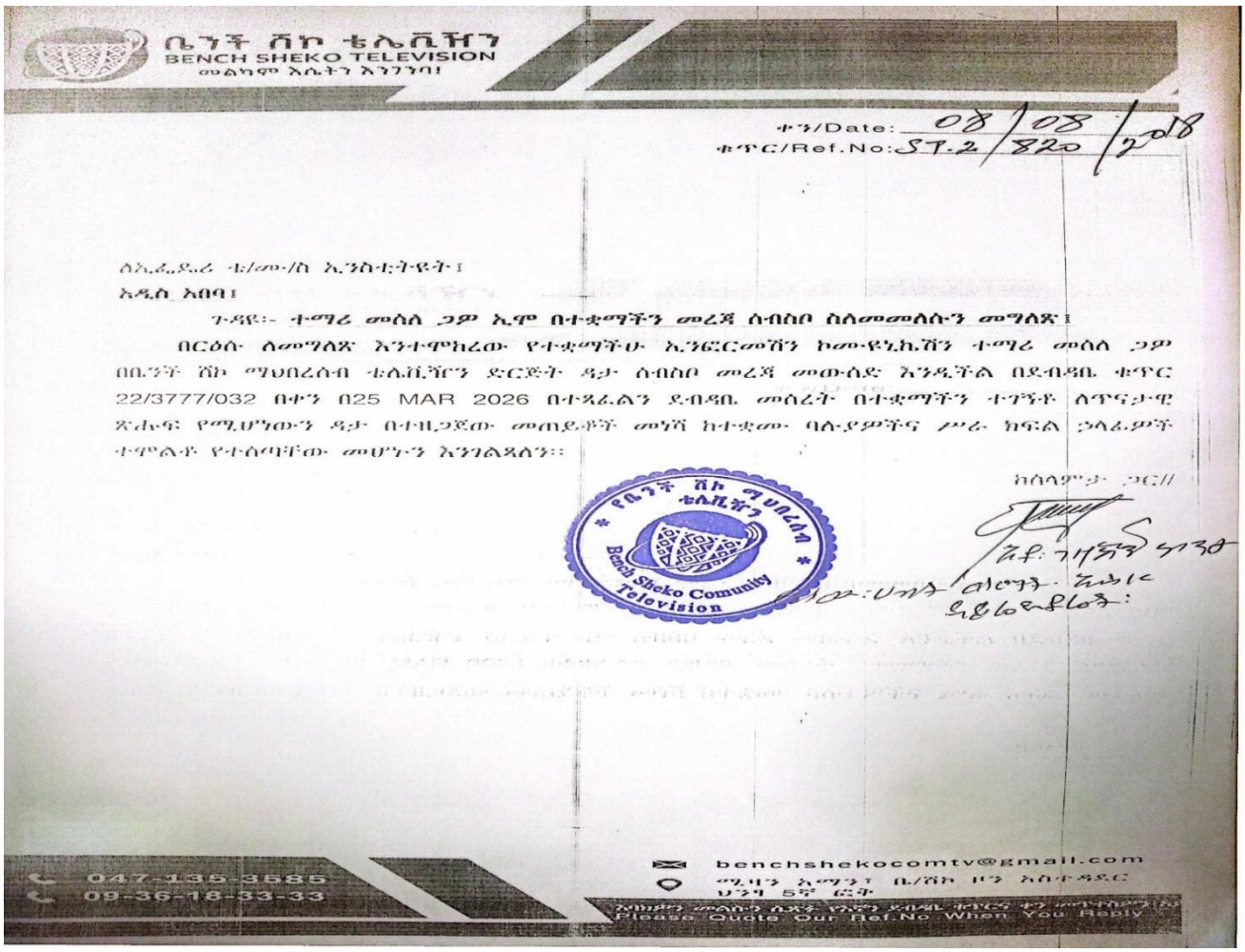
Amharic Sentence

የሚዛን አማን ከተማ አስተዳደር በ2018 ዓ/ም በሁለቱም ክፍለ ከተሞቹ ውስጥ የመንገድ ልማት ፕሮጀክቶችን እና የኮሪደር ልማት ሥራውን እንዲሁም የህዝብ አገልግሎት ሥራውን በተሻለ በፍጥነት በጥራት የከተማዉን ማህበረሰብ በማስተባበር ትኩረት ሰጥቶ እየሰራ ነው።

Sample chunked text Output

Word	POS Tags	Chunk Tags
የሚዛን	Noun	P B-NP
አማን	Noun	I-NP
ከተማ	Noun	I-NP
አስተዳደር	Noun	I-NP
በ2018	NUMCR	B-NP
ዓ/ም	Noun	I-NP
በሁለቱም	Adjective	I-NP
ክፍለ	Noun	I-NP
ከተሞቹ	Noun	I-NP
ውስጥ	Preposition	O
የመንገድ	Noun Phrase	B-NP
ልማት	Noun	I-NP
ፕሮጀክቶችን	Noun	I-NP
እና	Conjunction	O
የኮሪደር	Noun Phrase	B-NP
ልማት	Noun	I-NP
ሥራውን	Noun	I-NP
እንዲሁም	Conjunction	O
የህዝብ	Noun Phrase	B-NP
አገልግሎት	Noun	I-NP
ሥራውን	Noun	I-NP
በተሻለ	Adjective	B-NP
በፍጥነት	Noun	I-NP
በጥራት	Noun Phrase	B-NP
የከተማዉን	Noun Phrase	I-NP
ማህበረሰብ	Noun	I-NP
በማስተባበር	verb	B-VP
ትኩረት	Noun	I-VP
ሰጥቶ	Verb	I-VP
እየሰራ	Verb	I-VP
ነው	AUX	I-VP
።	Punctuation	O

Appendix 6. Data Collection Letter from Bench Sheko Community Television Service



ቤንች ሸኮ ቴሌቪዥን
BENCH SHEKO TELEVISION
መልካም አሴትን እንገንባ!

ቁን/Date: 08/08/2018
ቁጥር/Ref.No: ST.2/820/2018

ለኢፌዴሪ ቱ/መ/ስ ኢንስቲትዩት፣
አዲስ አበባ፣

ጉዳዩ፡- ተማሪ መስለ ጋም ኢሞ በተቋማችን መረጃ ሰብስቦ ስለመመለሱን መግለጽ፣

በርዕሱ ለመግለጽ እንተሞከረው የተቋማችሁ ኢንፎርመሽን ኮሙዩኒኬሽን ተማሪ መስለ ጋም በቤንች ሸኮ ማህበረሰብ ቴሌቪዥን ድርጅት ዳታ ሰብስቦ መረጃ መውሰድ እንዲችል በደብዳቤ ቁጥር 22/3777/032 በቀን በ25 MAR 2026 በተጻፈልን ደብዳቤ መሰረት በተቋማችን ተገኝቶ ለጥናታዊ ጽሑፍ የሚሆነውን ዳታ በተዘጋጀው መጠይቆች መነሻ ከተቋሙ ባለ-ደምችና ሥራ ክፍል ኃላፊዎች ተሞልቶ የተሰጣቸው መሆኑን እንገልጻለን።

ከሰላምታ ጋር//

ስልክ፡ 047 135 3585
09-36-18-83-33

benchshekocomtv@gmail.com
ሚዛን አማን፣ ቤ/ሸኮ ዞን አስተዳደር
ህንፃ 5ኛ ፎቅ

Please Quote Our Ref.No When You Reply

Figure 6.1. Data collection feedback letter