Addis Ababa University

Department of Linguistics

MSc. In Computational Linguistics

A Model for Amharic Idiom Identification Using Deep Learning

By: Sara Bihonegn

A Thesis submitted to the Department of Linguistics in Partial Fulfillment for the Degree of Master of Science in Computational Linguistic

Addis Ababa, Ethiopia
Jan 2025

Addis Ababa University

Department of Linguistics

Sara Bihonegn Kassaye

This is to certify that the thesis prepared by Sara Bihonegn, titled: A Model for Amharic Idiom Identification Using Deep Learning, submitted in partial fulfillment for the degree of Master of Science in Computational Linguistics, complies with the university's regulations and meets the accepted standards concerning originality and quality.

Signed by the examining committee:

Name	Signature	Date
Advisor: Demeke Asres (PhD)	_____	_____
Examiner: Derib Ado (PhD)	_____	_____
Examiner: Fitsum Assamnew (PhD)	_____	_____

Declaration

I, Sara Bihonegn declare that this research is my original work and has not been previously submitted for any other degree or publication. I acknowledge the contributions of all collaborators and sources used in this research.

Abstract

This work explores the detection of Amharic idioms using a hybrid machine learning and deep learning approach. Amharic, a Semitic language spoken in Ethiopia, has a large idiomatic vocabulary, making it challenging to perform natural language processing tasks.

We investigate the effectiveness of traditional machine learning algorithms, including Support Vector Machines (SVM), and Gradient Boosting, for idiom detection. Furthermore, we develop the potential of recurrent neural networks (RNNs), specifically Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM), for capturing the sequential nature of language and enhancing idiom identification accuracy. Our research aims to contribute to advancing Amharic natural language processing by developing robust and efficient idiom identification models. Experimental results on a curated Amharic idiom dataset were presented, the performance of the different algorithms was compared, and their strengths and weaknesses were analyzed. To measure the model's performance, we used accuracy, precision, recall, and F- score. The experimental results from the idiom identification indicate that the combination of SVM with Bi-LSTM, gradient boosting with Bi-LSTM, and Bi-LSTM alone achieved accuracies of 98.27%, 98%, and 98.18%, respectively. This study provides insights into the suitability of various machine-learning approaches for Amharic idiom identification and lays the groundwork for future research in this domain.

Keywords: Amharic idiom identification, Amharic idiom, and literal dataset, Deep Learning, Bi-LSTM, LSTM Machine Learning.

Acknowledgments

First and foremost, I sincerely thank the Almighty God and St. Mary for giving me the strength to complete this thesis.

I extend my heartfelt thanks to my esteemed advisor Demeke Asres (Ph.D.), for his invaluable mentorship and unwavering support. His dedication to my academic growth has been truly remarkable.

I am deeply indebted to Bezayt Yewondwossen for her exceptional support, encouragement, and insightful suggestions. Her willingness to share her expertise and offer constructive feedback has been invaluable in refining my research and ensuring its rigor. Her unwavering belief in my potential has been a constant source of motivation.

My heartfelt gratitude goes to my loving family for their support, unconditional love, and strong patience throughout my academic pursuits. Their love has been a constant source of strength and inspiration.

Thanks to the Tourism Training Institute for providing me with the exceptional opportunity to pursue my Master's degree. Their commitment to education and their dedication to fostering academic excellence have been instrumental in my growth as a scholar. This thesis is a testament to the collective support and guidance I have received from these individuals and institutions. Their contributions have been invaluable, and I am eternally grateful for their strong belief in my abilities.

Table of Contents

List of Table Page

List of Figure page

Acronyms and Abbreviation

NLP	Natural Language Processing
AI	Artificial Intelligence
Bi-LSTM	Bidirectional Long Short-Term Memory
LSTM	Long Short-Term Memory
DL	Deep Learning
IE	Idiomatic Expressions
ML	Machine Learning
SVM	Support Vector Machine
TF-IDF	Term frequency-inverse document frequency
RFC	Random Forest Classifier
API	Application Programming Interface
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
SMT	Statistical Machine Translation
GB	Gradient Boosting

CHAPTER 1 INTRODUCTION

1.1. BACKGROUND

Amharic is the federal working language of Ethiopia and is spoken by over 25 million people. According to Amharic Wikipedia, it's a Semitic language, which means it's related to other languages like Arabic and Hebrew. Amharic uses a unique script with 33 introductory characters, each with seven forms depending on the vowel sound. The Amharic script, known as Ge'ez or Ethiopic, consists of 33 consonantal characters, each of which can be modified by different vowel sounds. Here's an example using the base character "ሀ" (ha), which can be modified to represent different vowels' ሀ (ha) ሁ (hu) ሂ (hi) ሃ (ha) ሄ (he) ህ (h) ሆ (ho).

Idioms are expressions whose meanings are not directly inferable from the individual words. They enrich language by adding color and depth to communication, (Mäntylä, n.d.) .Idiom expression is a term often used or has a meaning different from the literal meaning of the words that make it up. Figurative language is used in idiomatic expression, which is one type of expression. Idiomatic phrases are collections of words with a shared meaning that is not connected to the meanings of the individual words. Idioms cannot be understood immediately from the term from which they are derived. Idiom expressions play a significant part in everyday speech and are essential factors in all languages. Expressions cannot be inferred immediately from the word they're deduced. Expressions cannot always be understood by simply looking at the words they're constructed with; rather, people may interpret them differently. Expressions are a regular part of all languages and a common part of our daily discussion. (Hinkel, 2017)

Understanding the language and the processes that underpin the automation of natural language processing (NLP) expressions is necessary to distinguish idioms from literals. Since idioms make up a significant portion of a language, it is crucial to build a model and algorithm for recognizing them to advance NLP-related research. Natural Language Processing (NLP) is a branch of computer science, particularly artificial intelligence (AI), concerned with enabling computers to interpret and reuse human language. The primary thing of NLP is to program computers to dissect and interpret large quantities of natural language data (Shahzad et al., 2024). Like numerous languages, Amharic indications have private expressions that draw from the country's history, myth, and artistic practices. For illustration, the expression "የቀን አንበሳ" (yak'an anbessa), literally means lion of the day, but it is used to relate to a person of great courage and strength. Idiom expressions are integral to the Amharic language, adding tone, emphasis, and artistic environment to everyday communication. Still, the nonliteral or hidden meaning of these expressions poses unique challenges for natural language processing (NLP) systems. Amharic language

comprehension skills are necessary for content analysis, dialogue systems, and machine restatement. These chops need the accurate identification and interpretation of private expressions. One major problem is to develop automated systems that can fete and understand similar private expressions in Amharic literature. Because typical NLP capture relies on rule-based or statistical models, it may not be able to capture the complex, environment-dependent nature of private language. Deep learning can significantly enhance the understanding and generation of idiomatic expressions in several ways. This paper provides an overview of the crucial rudiments and ways for creating an automatic system for receiving private expressions in the Amharic language. This study used a supervised machine learning approach to construct an idiom identification model for the Amharic language.

1.2. STATEMENT OF THE PROBLEM

The majority of Ethiopian languages, including Amharic idioms, are still in need of collection and organization. The majority of Amharic idioms and their explanations are kept manually (on paper), making it challenging to access and utilize them readily in digital format. There are far too many idiomatic terms utilized by the writers of various Amharic literature works. For instance, idioms can be found in FIKIR ESKEMEQABIR (ፍቅር እስከመቃብር), a well-known work of Amharic fiction.(Haddis Alemayehu, 1996)There are always a ton of idiomatic expressions in fiction that readers encounter, but they only understand them in the context of the text since they aren't given enough chances to identify idioms from the text using digitally gathered and arranged materials. The nature of idioms has an impact on other NLP investigations, such as semantic analysis, machine translation, and sentiment analysis. Another property of idioms that makes them challenging to comprehend for NLP systems is that they have both idiomatic and literal (nonidiomatic) uses. Statistical Machine Translation (SMT) systems, such as Google Translate, are one of the most important NLP applications that are badly affected by idioms. As a result, these systems are restricted to the direct translation of phrases that lack any syntactic or semantic background. An algorithm might not understand the idiom's figurative meaning if it takes each word at face value. Advanced NLP systems use context and a database of known idioms to correctly interpret these expressions. In the case of Amharic idioms or any other language’s idioms, incorporating cultural and linguistic context is crucial for accurate semantic interpretation. This often requires additional layers of processing and understanding of local tones, which can be a complex task for AI systems not specifically trained.

The other NLP research affected by idioms is sentiment analysis. Sentiment analysis is the most common text classification tool that analyzes an incoming text and tells whether the underlying sentiment is positive, negative, or neutral. Most sentiment analysis works by looking at words in

isolation, giving positive points for positive words and negative points for negative words, and then summing up these points. The sentiment analysis technology classifies a given text as negative if there is a negative word in the text. In Amharic, most of the time the negative word is formed from the prefix አል, አይ, አት…+ Root word + postfix ም,ችም Example አልበላም (አል+በላ +ም) to mean he has not eaten አትጠጣም (አት+ጠጣ+ም) to mean she will not drink አልሄደችም (አል +ሄደ+ችም) she has not gone. ልበአቁስል አይደለችም (she is not noisy). From the above examples, the first four sentences are classified as negative, due to the word, (አል, አይ, አት) the last sentence is positive, due to the given word, አይደለችም, and the last sentence is classified as negative based on considering the word, but due to the idiomatic phrase, the classification is false. Semantic analysis is the other NLP research affected by idioms.(Abebe Fenta & Gebeyehu, 2023)

The semantic analysis of natural language content starts by reading all of the words in the content to capture the real meaning of any text. Semantic technology processes the logical structure of sentences to identify the most relevant elements in the text and understand the topic discussed. The Idiom: "አንድ እጅ አያጨበጭብም" ("One hand cannot clap.") Literal Meaning: This sentence means that a single hand cannot create the sound of clapping. Figurative Meaning: This idiom signifies that cooperation is essential for success. It implies that teamwork and collaboration are crucial for achieving goals. Hence, we are going to develop an idiomatic identification model for Amharic texts by preparing a phrase-level dataset to enhance the model's performance since the phrase is the base for the document dataset.

This research is going to answer the following questions throughout and at the end of the research.

- How do we identify idioms from literals for the Amharic language?
- Which deep learning algorithm produces the best results in terms of performance?

1.3. MOTIVATION OF THE STUDY

Amharic idioms are deeply rooted in cultural, historical, and social contexts. One of the primary reasons for documenting these idioms is to preserve our cultural and linguistic heritage for future generations. Additionally, digitally compiling and safeguarding Amharic idioms is crucial for this preservation effort.

1.4. CONTRIBUTION OF THE STUDY

The contribution of this study is gathering different hard-copy idioms into soft copy. The study can provide valuable data and insights for the development of NLP. It contributes to machine translation, text analysis, and language understanding systems for the Amharic language.

Incorporating the understanding of Amharic idioms can improve the overall language proficiency of both native and non-native Amharic speakers.

1.5. OBJECTIVE

1.5.1. GENERAL OBJECTIVE

The main objective of this study is to develop a model for Amharic language idiomatic identification using deep learning.

1.5.2 SPECIFIC OBJECTIVE

- ✞ develop the idiomatic identification model using deep learning.
- ✞ Prepare a phrase-level dataset for idiomatic expression in Amharic.
- ✞ To determine the best model for idiomatic identification.
- ✞ Evaluate the performance of the proposed algorithm for idiomatic identification.

1.5.3. SIGNIFICANCE OF THE STUDY

This research will help to improve Amharic instruction, especially for non-native speakers. Better teaching resources and materials are being produced for the Amharic language. Amharic language processing using natural language processing and computational linguistics enhances the activities; this includes machine translation, sentiment analysis, and text summarization. expanding the potential of Amharic-based artificial intelligence and natural language comprehension systems. supporting the development of Amharic-based services and apps.

Idiomatic phrases are frequently employed in published Amharic fiction books and other publications, according to writers (DagnachewAmsalu, 1993) to grab the reader's attention and effectively communicate a point. As an illustration, the idiomatic phrase "እሳት ሆኗል" (meaning "it is a fire") can be articulated as either "ተወዷል" (expensive) or "ዋጋ ጨምሯል" (high price). The First expression, however, has a greater impact than the second. Algorithms for identifying idiomatic expressions in Amharic are necessary to make concepts easier to understand when they are given in historical, educational, and fictional texts. Idioms often reflect the values, beliefs, and historical experiences of a culture. Understanding Amharic idioms provides insights into Ethiopian culture and societal norms. To address this, a model that correlates collections of Amharic idiomatic expressions with their idiomatic meanings must be developed. The reader might not be able to identify the sentences that are joined to make idioms or determine whether they even exist. Therefore, idiom identification is a very important application. The proposed model would clarify

idiomatic expressions and it can be incorporated into other NLP research to avoid the problems that happened to the existence of idiomatic expressions.

1.5.4. SCOPE OF THE STUDY

Amharic idioms might come from single words, sentences, or clauses. This analysis includes phrases and words from idiomatic expressions. Idiomatic identification is taken out of the given text and classified as literal and idiom, not including the hidden meaning. For our work, we used phrase-level idiomatic expression identification. The main problem with idiomatic phrases is that they might be used literally or in casual speech. Idiomatic expressions are characteristically pure or semi-pure. All idiomatic phrases, whether pure or semi-pure, will be used in our thesis. However, the type of substance pure or semi-pure is not considered in this study because it is challenging to distinguish them. Only text files are taken into account in this study; audio, video, and image files will not be included.

1.5.5. ORGANIZATION OF THE RESEARCH WORK

The thesis consists of five main chapters including this chapter. The first chapter includes the introduction, statement of the problem, motivation, objective, significance, and scope of the study. Chapter two covers the overview of the Amharic language, writing system, punctuation marks, numbers, and the related works that have been done before in idiomatic expression identification. The third chapter discusses the research methodology used, approaches, and tools used to develop the proposed model. The fourth chapter describes how to custom deep learning approach for idiomatic expression identification from Amharic texts. By comparing different selected algorithms with experiments and evaluating their performance. Finally, the conclusion, recommendation, and some future works have been presented in chapter five.

CHAPTER 2 LITERATURE REVIEW

2.1. INTRODUCTION

Amharic idiom is a crucial task for various natural language processing (NLP) applications, including machine translation, sentiment analysis, and text summarization. However, the research on Amharic idiom identification is still in its early stages, with limited resources and studies available.

This literature review explores the existing research on Amharic idiom identification, the lack of annotated corpora specifically for Amharic idiom identification is a major problem. Existing corpora are often general-purpose and lack the necessary annotations for idioms. While some lexical resources for Amharic exist, they are often incomplete or lack specific information about idioms. Lack of Standardized Definitions, there is no universally accepted definition of what constitutes an idiom in Amharic, leading to inconsistencies in identification and annotation. Machine learning and deep learning techniques are key in idiom detection for natural language processing applications. Amharic, the Ethiopian language, is rich in idiomatic terms, crucial for literature and communication, but presents challenges for computer linguistics due to its complexity.(Endalie et al., 2023)

Traditional Approaches to Idiom Identification (Rule-Based Methods): Rule-based methods were initially employed to discover idioms in languages like Amharic, but are not scalable or flexible enough to accommodate new idioms. Rule-based idiom identification is a method used in natural language processing (NLP) and computational linguistics to detect idiomatic expressions in text based on predefined linguistic rules.

Statistical Methods: The statistical method approach for Amharic idiom identification involves using quantitative techniques and data-driven models to detect idiomatic expressions in the language.Large corpora have been analyzed using statistical techniques to identify idioms. Methods like frequency analysis and n-grams can assist in identifying idiomatic usage patterns. However, these approaches frequently fail to capture the polysemy and contextual variances in the idiomatic expression.

Machine Learning Approaches: The machine learning approach for Amharic idiom identification involves using algorithms that learn from data to recognize idiomatic expressions. Machine learning techniques can be employed to classify idiomatic expressions. Feature extraction methods like Bag of Words, TF-IDF, or more advanced embeddings (Word2Vec or BERT) can be utilized.

Supervised learning models (like SVM, Random Forest, or Neural Networks) can be trained on labeled datasets containing idioms and non-idioms.

Word Embeddings: Word embedding is a technique used in natural language processing (NLP) to represent words as continuous vectors in a high-dimensional space. This approach captures semantic meanings and relationships between words, making it particularly useful for tasks like idiom identification in languages such as Amharic. The development of word embeddings has revolutionized NLP, providing a mechanism to record semantic links in language. Amharic word vector representations have been created using methods like Word2Vec and Glove. By encapsulating contextual similarities and semantic interpretations, these embeddings can aid in identifying idioms.(Melamud et al., n.d.)

Deep Learning Approaches Recurrent Neural Networks (RNNs): Recurrent Neural Networks (RNNs) are a class of artificial neural networks designed to process sequential data, making them particularly useful for tasks like Amharic idiom identification.Long Short-Term Memory (LSTM) networks in particular, which are RNNs, have demonstrated potential in sequence-based tasks such as idiom recognition. LSTMs can learn contextual dependencies, which are essential for comprehending idiomatic expressions, by encoding the sequential character of language (Kuncoro et al., n.d.).

Transformer Models: The Transformer model is a powerful architecture in natural language processing (NLP) that has significantly advanced the field, particularly in tasks like idiom identification. By utilizing attention mechanisms, the emergence of transformer designs, including BERT and GPT, has greatly boosted NLP jobs. These models are appropriate for idiom identification since they have shown remarkable ability in comprehending context and semantics. Transformer models may be used to identify Amharic idioms more accurately, according to recent research that has started to investigate this possibility.

Challenges and Future Directions: Despite improvements in machine learning and deep learning techniques, idiom detection remains a difficult problem. A major barrier to creating workable models for Amharic idioms is the lack of annotated materials. Amharic has a rich morphological structure, with prefixes, suffixes, and infixes that can complicate the identification of idioms. Variations in word forms can lead to difficulties in recognizing idiomatic expressions. The evolving nature of language requires continuous research and regular model retraining, as idioms can change over time.

This chapter aims to demonstrate the most important aspects of existing works, and theoretical and methodological contributions relevant to idiom identification. The overview of the Amharic language, its grammatical nature, morphology, the writing system in the Amharic language was

discussed in this chapter. A study of similar works of literature was reviewed to establish the model for this research as well as to organize the research concept. The literature on idiom recognition and idiom properties has been reviewed specifically for this study.

2.2. OVERVIEW OF AMHARIC LANGUAGE

The 1995 Constitution of Ethiopia accords equal state recognition to all Ethiopian languages, designates Amharic as the working language of the federal government, and allows members of the federation to determine by law their respective working languages.; the majority of national news and papers are written in this language. Still, few online electronic resources are available for the Amharic language, and not much development has gone into creating various computer-based apps. It was written using the Fidel (ፊደል) or abugida script, which was taken from the extinct Ge'ez language. (Fissaha & Haller, n.d.)

2.2.1. AMHARIC MORPHOLOGY

Amharic morphology, the study of the structure of words in the Amharic language, plays a vital role in various machine learning applications, especially in natural language processing (NLP). Incorporating Amharic morphology into machine learning models enhances their understanding of the language's unique features, leading to improved performance in NLP tasks. By focusing on morphological structures, models can better handle the complexity of word formation in Amharic. Amharic is a Semitic language and uses the Ethiopian alphabet for writing. The writing system has 33 base characters which change their shapes into seven different forms due to vowels. Due to its Semitic characteristics, Amharic has a rich and complex morphological structure. According to (Assabie, july,2021), thousands of surface words can be generated from a base form.

Amharic words are inflected for person (first, second, third), gender (feminine, masculine), number (singular, plural), case (subjective, objective, possessive), definiteness (definite, indefinite), tense (past, present, future), aspect (perfective, imperfective), politeness (impolite, polite), etc. This is achieved by adding prefixes, infixes, and suffixes. The grammatical relations like subject, object, and syntactic information might be indicated morphologically at the word level. For instance, the word ይሰጣታል /jɨsəʈatalɨ ‘he will give her’/ is composed of the subject marker for imperfect tense ይ…ኣል /jɨ…ʔalɨ/, imperfect verbal stem ሰጥ /səʈɨ/ and the object marker ኣት /ʔatɨ/. Verbs exist in perfective, imperfective, jussive, gerund, and infinitive forms. Each form has its stem template. Stems of verbs can be classified as basic and derived stems. The verbs' basic stems are modified internally (by inserting infix) or externally (by attaching prefixes) to form derived stems.

The derived stems include passive, causative, infinitive, and reduplicative. Passive stems are formed by attaching the prefix 'ተ /tə/' on basic stems, causative stems are derived variably by attaching the prefix 'አ /ʔə/', 'አስ /ʔəsɨ/', or 'አት /ʔətɨ/', and infinitive stems are formed by adding the prefix 'መ /mə/' on basic stems. Moreover, derived stems can be formed by reduplicating the character of a basic stem. Both types of stems may be preceded by many prefixes and followed by many suffixes. The orthography of the Amharic language can combine one or more functional words and inflectional morphemes. Amharic morphemes play significant roles both in morphology and syntax. Most Amharic words are composed of a basic form and many attached affixes. Prefixes can be the prepositions (ከ /kə/, በ /bə/, etc.) or genitive (የ /jə/), negations (አል /ʔəlɨ/), and conjunctions (እንደ /ʔɨnɨdə/, etc.) while suffixes include plural marker (አች /ʔotʃɨ/ or ዎች /wotʃɨ/), possessive (ኤ /ʔe/, ኡ /ʔu/, አችን /ʔətʃɨnɨ/, ሽ /ʃɨ/, etc.), or a definite marker (ኡ /ʔu/, ው /wɨ/, ዋ /wa/), and connectors (ና /na/). Gender, number, case, and definite markers can be suffixed to the stem of nouns. The sets {ኢት /ʔitɨ/, ዋ /wa/, ኡ /ʔu/, ው /wɨ/}, {አች /ʔotʃɨ/, ዎች /wotʃɨ/, አን /ʔənɨ/, እየ /ʔɨjə/, አት /ʔətɨ/}, {ን /nɨ/, ዬ /je/, ኤ /ʔe/, ዎ /wo/, ህ /hɨ/, ሽ /ʃɨ/, ኡ /ʔu/, ዋ /wa/, አችሁ /ʔətʃɨhu/, አችን /ʔətʃɨnɨ/, አቸው /ʔətʃəwɨ/}, and {ኡ /ʔu/, ዋ /wa/, ው /wɨ/, ኢቱ /ʔitu/, ይቱ /jɨtu/} are gender, number, case, and definite markers, respectively.

The most common suffixes to derive adjectives are አማ /ʔəma/ and አዊ /ʔəwi/ from nouns, and ኢ /ʔi/ from verbs. The language has several lexical variations and clitics. Sometimes, there is no clear demarcation between clitics and content words in the orthography. The clitics such as prepositions and conjunctions, which have syntactic roles, indicate grammatical relations with the content words. An Amharic content word can represent a phrase, a clause, or a sentence.

From a computational point of view, segmenting a word into its morphemes is very crucial in many Amharic IR and NLP applications. For instance, Amharic IR systems require words in documents and queries to be segmented correctly into their stems, roots, and affixes. However, separating morphemes from surface words is a challenging task. This problem harms the performance of different applications as it results in vocabulary mismatch problems for words generated from the same root form. Therefore, Amharic raw text needs to be morphologically analyzed to get the desired results.

2.2.2. AMHARIC WORD CLASSES

In Amharic, words can be categorized into various classes based on grammatical functions. Understanding these word classes is crucial for natural language processing (NLP) and machine learning applications. Amharic words are categorized under six basic classes, namely, ስም (noun), ተዉላጠ ስም (pronoun), ግስ (verb), ቅፅል (adjective), ተውሳከ ግስ (Adverb), and መስተዋድድ (preposition)

based on morphology and position of the word in Amharic sentences. (ይማም, (1987)).

2.2.3. NOUN (ስም)

Nouns in Amharic can be either simple or compound (Eg. “ሰማይ” - “sky“, “ውሃ“ -“water”, and “እሳት“

“fire“). Compound words can be used to make nouns (sometimes by affixing the vowels ኧ and ኦ): Noun (ብረት) + Noun (ድስት) => ብረት ድስት; Noun (ልብ) + Verbal (ወለድ) => ልብወለድ; from Nouns by suffixing bound morphemes Nouns can be derived (Example: Noun (መንገድ) + morpheme (ኛኛ) => መንገድኛኛ=>መንገደኛ).

In the Amharic Language noun class, there are two gender indicators called masculine and feminine. Masculine means male gender indicator word, and feminine means female gender indicator. However, for things that are not naturally male or female, the gender tends to be used when the entity is small or adorable; the gender is female otherwise used as male. The feminine gender suffix (-it or yt, phonologically conditioned) is accustomed to marking femininess in otherwise masculine cases. In addition to this expansion, Amharic nouns can also be expanded in number to form a plural. Consider the following example: [ካህን - ካህናት], [መምህር - መምህራን], [ልጅ- ልጆች], [ክፉ - ክፉዎች], [መነኩሴ - መነኮሳት].

2.2.4. PRONOUN (ተውላጣ ስም)

Amharic has a rich morphology, where pronouns change form based on case, gender, and number. This complexity can pose challenges for models trained on languages with simpler structures, requiring specialized tokenization and embedding strategies. Pronouns serve to replace nouns and avoid repetition. Here’s a breakdown of the different types of pronouns in Amharic: The following are some of the pronouns in Amharic እሱ, እስዋ, እኔ, አንተ, አንች…; quantitative specifiers, which includes አንድ, አንዳንድ, and possession specifiers such as የእኔ, የአንተ, የእሱ,የእሷ, የነሱ...Pronouns in Amharic must agree in gender and number with the nouns they replace. Understanding the context is essential for appropriate usage, especially with personal and possessive pronouns. Incorporating Amharic pronouns into machine learning systems enhances their ability to understand and generate human-like text, making applications more effective in real-world scenarios.

2.2.5 VERB (ግስ)

Amharic verb formation involves various patterns based on roots, conjugation, aspect, voice, and derivation. The complexity of these patterns reflects the rich morphological structure of the

language, making it essential for learners and linguists to understand them for effective communication and analysis.

Amharic, the formation of verbs involves various patterns, including the root and specific suffixes or prefixes that indicate tense, aspect, mood, and subject. Any word that can be used at the end of a sentence and accepts suffixes such as /ሀ/, /ሁ/, /ሸ/, etc. in the Amharic language. Verbs are derived from verbal roots by affixing the vowel letter. Example: ስ -ብ -ር ስኧብኧር [ሰበር -] may derive from compound words of stems and verbs. Example: - ስብር + አለ= ስብር አለ, ፀጥ+አደረገ =ፀጥ አደረገ.

2.2.6. ADJECTIVE (ቅፅል)

In Amharic, adjectives describe or modify nouns, providing information about qualities, quantities, or characteristics. Adjectives in Amharic typically follow the nouns they modify. They can agree with

the noun in gender (masculine or feminine) and number (singular or plural). a word that describes a person or thing, for example, 'big', (ትልቅ), ('red'(ቀይ) and 'clever' (ፈጣን) in a big house, red wine, and a clever idea." An adjective is "a word belonging to one of the major form classes in any of numerous languages and typically serving as a modifier of a noun to denote a quality of the thing named, to indicate its quantity or extent, or to specify a thing as distinct from something else," according to the Merriam-Webster Dictionary. Accordingly, words that qualify a noun or an adverb that occurs before a noun, e.g., ፈጣን ልጅ, and after an adverb (በጣም ፈጣን). When pluralizing an adjective, it will repeat the previous letter of the word's last letter, e.g., ረዥም=> ረገዥም, አጭር =>አጫጭር, ጥቁር =>ጥቋቁር, etc. By infixing vowels between consonants, adjectives can be formed from verbal roots.

2.2.7. ADVERB (ተውሳከ ግስ)

An adverb is a word that can modify or describe a verb, adjective, another adverb, or entire sentence. Adverbs can be used to show manner (how something happens), degree (to what extent), place (where), and time (when). Incorporating Amharic adverbs into machine learning applications, especially in natural language processing (NLP), can enhance the understanding and interpretation of text. Amharic language has a rich set of adverbs that can modify verbs, adjectives, and other parts of speech. In Amharic Language, Adverbs qualify verbs by adding additional concepts to the sentence. Example: - ገና, ዛሬ, ቶሎ, ምንኛ, ክፉኛ, እንደገና, ልግምኛ, etc. (Baye Yimam, 1999)

2.2.8. PREPOSITION (መስተዋድድ)

Amharic prepositions are used to indicate a relationship between words in a sentence. A preposition is a word that can be used before a noun to conduct adverbial actions such as place, time, cause, and so on, but it can't take any suffix or prefix from the beginning to the end of the character and can't be used to make a new word. It consists of ከ ፣ ለ ፣ወደ ፣ ስለ ፣ ላ ይ ፥ በ፥ እንደ እስከ፥ጋር etc. these prepositions could be beneficial for training models on understanding Amharic language structure.

2.2.9. AMHARIC PHRASES (ሐረግ)

Phrases, expressions, idioms, saying, and locutions all refer to grammatically related groups of words. A phrase is a sequence of two or more words that make up a grammatical construction, usually lacking a finite verb and hence not a complete clause or sentence. A phrase (ሐረግ) is a collection of words that convey some meanings but do not make complete sense. It is always a part of a sentence and group of words, often carrying a special idiomatic meaning; in this sense, it is synonymous with expression. In linguistic analysis, a phrase is a crowd of words (possibly a solo word) such as "ና" that

functions as essential in arranging a sentence, a single unit within a grammatical hierarchy.

2.2.10. AMHARIC SENTENCE (አረፍተ ነገር)

A sentence is a collection of words that expresses or conveys a complete meaning. In most Amharic sentences, words follow a subject-object-verb pattern; English usually follows a subject-verb-object pattern. The order of words inside Amharic sentences is different than in English. Generally, the verb goes after the sentence and the structure is Subject / Object / Verb. It can be a declaration used to announce, clarify, or argue an occasion (David A. Odden Amharic Syntax: A Generative Perspective). Example Ayele [subject] eats [verb] his lunch [object], In Amharic አየለ [subject] ምሳዉን [object] በላ [verb] as a phrase to express anything, a sentence may be incomplete. The following are two types of Amharic sentences: - sentences, both basic and complicated. A simple sentence has only one verb within a phrase. A complex sentence is categorized as a simple sentence and formed by complex phrases. The subject-object-verb arrangement is common in Amharic sentences (SOV). However, OSV sentences do occur from time to time. For example: - the sentence 'ማርታ በቀለን መታችዉ"Marta Bekelen Metachwu, which is in SOV order, can also be written as 'በቀለን ማርታ መታችዉ " Bekelen marta Metachwu " in OSV order. Unless the word (sentence object)

includes the object marker, the meaning of a phrase might be changed depending on where words are placed in the sentence. „ን"/"-n". For instance, „ጅቡ ውሻ ይበላል" /"jib wusha ybelal" and 'ውሻ ጅቡ ይበላል" /"wusha jib ybelal": Both phrases utilize the same words, but they have distinct meanings. „ጅቡ"

/"jib" and „ውሻ"/"wusha" are the subjects of the first and second sentences, respectively ((Baye Yimam, 1999). There are no subject markers or morphemes in Amharic nouns (affixes). However, a subject can be identified from its place in a sentence. „-ን" /"-n" is a suffix that is used as a sign for Amharic objects.

2.2.11. AMHARIC PUNCTUATION MARKS

Punctuation marks are marks indicating how a piece of written text should be read (silently or aloud) and, consequently, understood. The oldest known examples of punctuation marks were found in the Mesha Stele from the 9th century BC, consisting of points between the words and horizontal strokes between sections. The alphabet-based writing began with no spaces, no capitalization, no vowels (see abjad), and with only a few punctuation marks, as it was mostly aimed at recording business transactions. Only with the Greek playwrights (such as Euripides and Aristophanes) did the ends of sentences begin to be marked to help actors know when to pause during performances. Punctuation includes space between words and both obsolete and modern signs. By the 19th century, the punctuation marks were used hierarchically, according to their weight. Six marks, proposed in 1966 by the French author Hervé Bazin, could be seen as predecessors of emoticons and emojis.

In Amharic, there are many punctuation marks. There are approximately seventeen punctuation marks in the Amharic language writing system (Tewodros Hailemeskel, 2003)However, only a few of them are widely used. In Amharic, punctuation marks are similar to those used in English, but they may have some unique features. Here are some common punctuation marks used in Amharic:

- Hulet Neteb (፡) two square dots arranged like a colon: This is used to separate one Amharic word from the other in the Amharic writing system. However, these days its function is replaced by spaces.
- Four square dots (።) arranged in a four-sided pattern is the basic one in the Amharic language which is used to represent the end of a sentence. It has the same use as a full stop in the English language.
- Netela sereze (፣) is an equivalent of a comma used for separate lists in Amharic text.
- Derib sereze (፤) which is the equivalent of a semi-colon, may also be found in use as a list separator. The Amharic writing system has also borrowed additional punctuation marks from

other foreign languages (?!, “, ”, „, /, \, etc.) Ge'ez numbers, used in the Ge'ez script (also known as Ethiopic), have their symbols. Here are the symbols for the numbers 1 to 10 in Ge'ez.

2.2.12. AMHARIC NUMBERS

Amharic numbers are unique and have their own script. The Amharic Number system writing has 20 single characters which represent one (1/፩ up to 9/ ፱), tenths (ten/፲ to ninety/፺), hundred (፻), and ten thousand (፼). These characters are Ge'ez numbers, used in the Ge'ez script (also known as Ethiopic), and have their symbols. Here are the symbols for the numbers 1 to 10 in Ge'ez:

2.2.13. OVERVIEW OF AMHARIC IDIOMS

Amharic is the working language of Ethiopia., is rich in idioms, which are expressions that have a figurative meaning different from the literal meaning of the words. These idioms add color and depth to the language, reflecting the culture and history of the people. Many idioms use animals to describe human behavior or situations. For example:"እንደ ውሻ መጮህ" (Enda wusha mech'oh) - "To bark like a dog": This idiom means to be loud and aggressive."እንደ ድመት መንቀሳቀስ" (Enda d'met menkesakes) - "To move like a cat": This idiom means to be stealthy and agile."እንደ አንበሳ መጮህ" (Enda anbesa mech'oh) - "To roar like a lion": This idiom means to be powerful and commanding."እንደ አሳ መዋኘት" (Enda asa magnet) - "To swim like a fish": This idiom means to be comfortable and at ease.

Amharic idioms often use food and drink to express emotions and situations. For example:"እንደ ማር ጣፋጭ" (Enda mar tafach) - "Sweet like honey": This idiom means to be very pleasant."እንደ ጨው መራራ" (Enda chew merara) - "Bitter like salt": This idiom means to be unpleasant or painful.Body parts are frequently used in Amharic idioms to describe actions or feelings. For example:"ልቡ ተሰበረ" (Lib'u tesebere) - "His heart broke": This idiom means to be heartbroken or deeply saddened."አይኑ አየ" (Ay'nu aye) - "His eyes saw": This idiom means to understand or realize something.

"እንደ ዝናብ መውረድ" (Enda zenab mewered) - "To fall like rain": This idiom means to be sudden and unexpected."እንደ ፀሐይ መብራት" (Enda t'sehay mebrat) - "To shine like the sun": This idiom means to be bright and radiant."እንደ ኮከብ መብረቅ" (Enda kokeb mebrek) - "To sparkle like a star": This idiom means to be beautiful and dazzling. Amharic Idiomatic expressions are phrases that cannot be interpreted by taking individual word meanings. It can be formed by a single term, or phrase level. For example: Single word: አለዘበ (“Alezebe”), ሰብቷል (“Sebtwal”) ለበለበ (“lebelebe”) ረከሰ (“rekese”). Phrases level: ሁለት ምላስ (“Hulet milas”), እጅ አጠረው (“Eji aterew”) ሹል አፍ (“shul

af") ጆሮጠቢ(jorotebi),ሆደ ሰፊ(hode sefi) , ልበ ቢስ (libe bis).

Idiomatic expressions are complicated and have a pure and semi-pure quality. This implies that the meanings of the idioms can be both literal and idiomatic. Semi-pure idiomatic expressions have two meanings: literal and idiomatic, while pure idiomatic expressions have just one meaning (idiomatic). For instance, the Amharic idiomatic phrase "እንቶ ፈንቶ" / "A hundred funnels" has only one possible interpretation: "ተራ፤ ዝባዝንኬ ነገር" / "Ordinary", "Something straying." Certain idioms can be interpreted both literally and idiomatically, in contrast to pure idiomatic terms. The Amharic idiomatic phrase "እጅ ሰባራ" / "A Broken Hand" can be taken literally to mean "የተሰበረ እጅ" "Broken hand," or it can be understood idiomatically as "የማይረባ ስራ የሰራ" / "Doing nonsense."Many types of research have been conducted research related to idioms in different languages and their effects were analyzed on language translation Different scholars do idiom identification using different methods like using meaning VNC part of tag sequence, sentential distribution word embedding (S Huet, G Gravier, P Sébillot. Computer Speech & Language)). Most work on the phrase classification stream imposes syntactic restrictions. Verb/Noun restriction is imposed and Preposition Noun-Verb restriction is imposed.

The most recent studies used word embedding models by vector representation of phrases through various methods, such as term frequency of phrases and the Word2Vec approach for idiom identification (Young et al. 2018) (Lavanya and Sasikala 2021). The research focused on the meaning of idiom terms so that the properties of individual words in a phrase vary from the properties of the phrase in itself. The success of the study was evaluated using a union and intersection methodology. As a result of this research, a model that recognizes idiomatic speech through dictionary-based type was developed. It takes VNC POS tag sequence only and is difficult for Amharic idioms because of the ambiguity of idioms like እጅ ሰጠ, የግንባር ስጋ,ሀሞተ ኮስታራ ,ልበ ገር …

The study was conducted by employing dictionaries to automatically identify idiomatic terms (Muzny & Zettlemoyer, n.d.). The study comprised five graph-based features and three lexical characteristics. The study's main goal was to extract English-language phrases from web data. The findings were evaluated using the Lesk word sense disambiguation algorithm and the Wiktionary default rule. A word becomes an idiom if its meaning is not clear, however, not all unclear words are idioms. In the study, the Lesk Word Sense Disambiguation algorithm was applied. Their word-matching algorithm was restricted to dictionary terms. (Peng et al., n.d.)

2.3. Gap analysis

Most previous research focuses on idiom identification in languages like Chinese and English, without adapting their methods and structure because almost all have the same structure except

Amharic and a few languages. It is essential to have a sizable, excellently documented corpus of idioms. There should be variety in the contextual and grammatical patterns within this corpus. Although several studies have produced idiomatic expression databases, the scope and variety of these datasets are still restricted.

The previous study used 200 labeled phrases for testing and 800 labeled expressions for training. This data size may not be sufficient for the deep learning model. It is necessary to better understand the linguistic characteristics that set Amharic idioms apart from other expressions. This requires separating idioms' pragmatic, syntactic, and semantic components. Creating efficient machine learning models is crucial, especially for Amharic idiom recognition. Investigating various algorithms and feature engineering strategies may be part of this. One issue is the inability to use a deep learning model due to insufficient data; the researcher addresses this by filling in the gaps and incorporating more data than before, to increase the model's efficiency in this work, the researcher employed deep learning techniques and ensemble machine learning.

2.3.1. Related work

No	Author	Title	Method used	Data size	Result
1	Peng and Feldman 2016	Automatic idiom recognition with word Embedding	Tf-idf, phrase-idf, phrase-tf-idf, CoVAr, context	2984 VNC	92%
2	Anduamlak Abebe Fenta. August 2023	Idiom Identification Model	supervised machine learning approach	1000 phrase	97.5%
3	Demeke.Endale December2023	Deep learningbased idiomatic expression	(CNN)with FastText embedding	3,300 phrase	80%
4	Alemayehu A. G. and colleagues	Idiom Recognition Model	CNN with Fast Text	1,700 idiom 1,600 non idiom	98%

TABLE 1 RELATED WORK

2.4. APPROACHES FOR IDIOM IDENTIFICATION

Idiom expression detection challenges can be approached using three different types of methods: rule-based, deep learning, and classical machine learning. To handle the semantic expression of the sentence, rule-based models rewrite the rules using logic-based rules (Muzny & Zettlemoyer, n.d.). These traditional models comprise the representation and judgment components. The meaning and context of the words are disregarded by rule-based approaches, also referred to as context-less identification strategies. Since the context in which words appear in the text influences whether or not an utterance is an idiom, it is difficult to identify idioms using this technique (Peng et al., n.d.).

2.4.1. CLASSICAL MACHINE LEARNING APPROACHES

Classical machine learning refers to traditional techniques and models used for building predictive systems before the rise of deep learning. These approaches typically involve statistical methods and are well-suited for various tasks in supervised and unsupervised learning. The study of computer algorithms that enable computer programs to automatically improve when the dataset is supervised that is, when the dataset consists of pairs of input objects (usually vectors) and a desired output value (referred to as the supervised signal), which can be a label or a number is known as machine learning, also known as narrow artificial intelligence (Shahzad et al., 2024). The majority of issues with rule-based approaches can be resolved by traditional machine learning techniques, which acquire knowledge of the structure and meaning of colloquial language.

2.4.2 SUPPORT VECTOR MACHINE

The support vector machine (SVM) is a statistical learning theory-based machine learning technique ((Bzdok, 2018))) A support vector machine constructs a hyperplane or a group of hyperplanes in a high or infinite dimensional space for classification. Any class's hyperplane with the largest distance to the closest training data point (functional margin) achieves good separation; in general, the higher the margin, the lower the classifier's generalization error. SVM employs a non-parametric technique that uses a binary classifier and is capable of processing a large amount of data quickly.

The kernel setting and hyperplane selection have an impact on accuracy and performance. The main benefits of SVM are that it removes the overfitting issue, offers greater freedom in threshold

form selection, and has a nonlinear transformation and strong generalization capacity. Complex calculations are minimized. Error frequency and decision rule complexity are easily controlled. Limited transparency, time-consuming training, an opaque algorithm structure, and trouble figuring out the ideal parameters in the presence of nonlinearly distinct training data are some of the drawbacks of SVM. SVM reduces the complexity of the decision rule and the calculation. Training speed in Support Vector Machines (SVM) is dependent on the quantity of learning data and class separation. SVM is more memory-efficient and computationally intensive when inferring the session of a new observation because it only needs a tiny portion of training data to generate the classification rule.

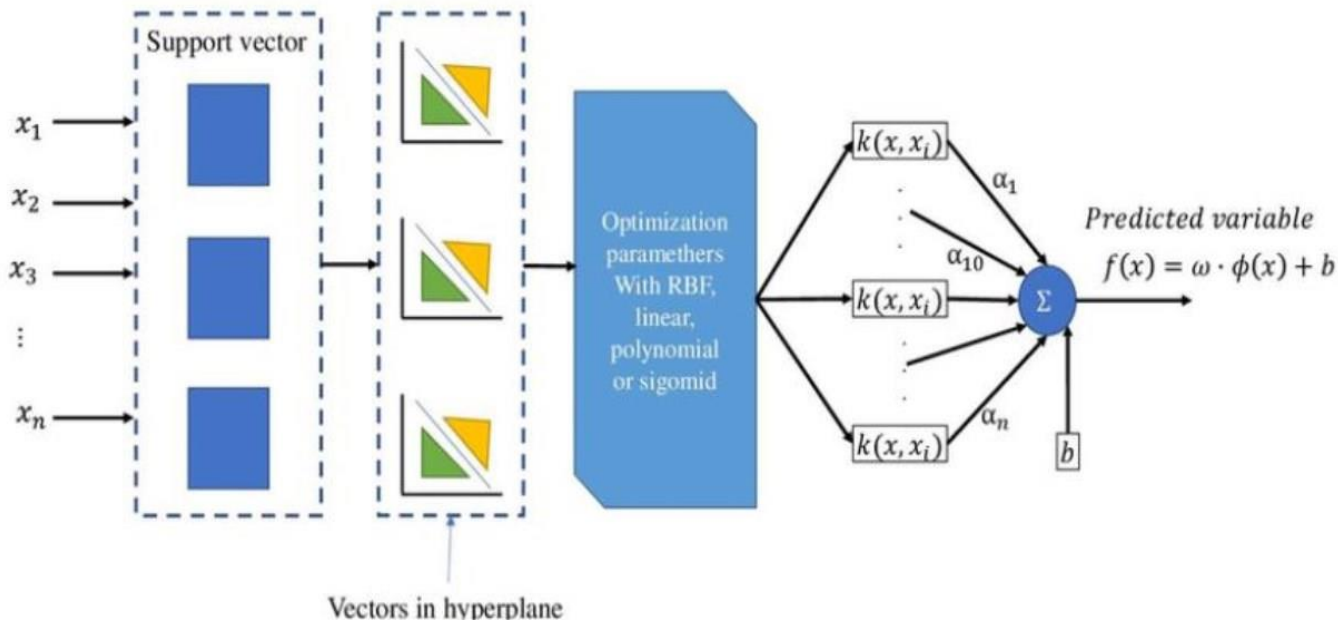


FIGURE 1 SVM MODEL ARCHITECTURE

2.4.3. Random Forest

Unpredictable Forest Classifiers Random forests are based on decision trees and are a modeling and behavior analysis tool. It is made up of numerous decision trees, each of which represents a different example of how the data supplied into the random forest is classified. According to, the random forest technique selects a prediction by weighing each instance separately and selecting the one with the plurality of votes. A decision tree set is produced by a random forest. Random Forest used the randomization approach to achieve variation amongst basic decision trees; this approach is compatible with bagging or random subspace techniques. The following actions need to be taken to create each tree in the random forest: If the training set has N records, N records are

randomly sampled but replaced by the original data. This is a bootstrap sample. This sample will be a training set for growing the tree.

2.4.4. GRADIENT BOOSTING

for both classification and regression applications, gradient boosting is a potent machine-learning method. To produce a strong, accurate model, it iteratively constructs an ensemble of weak learners (usually decision trees) and combines their predictions. Attains cutting-edge results on several datasets with regularity. less susceptible to data outliers than alternative algorithms. Manages intricate relationships: able to record intricate feature interactions. Every feature in the model can have its importance estimated by the algorithm.

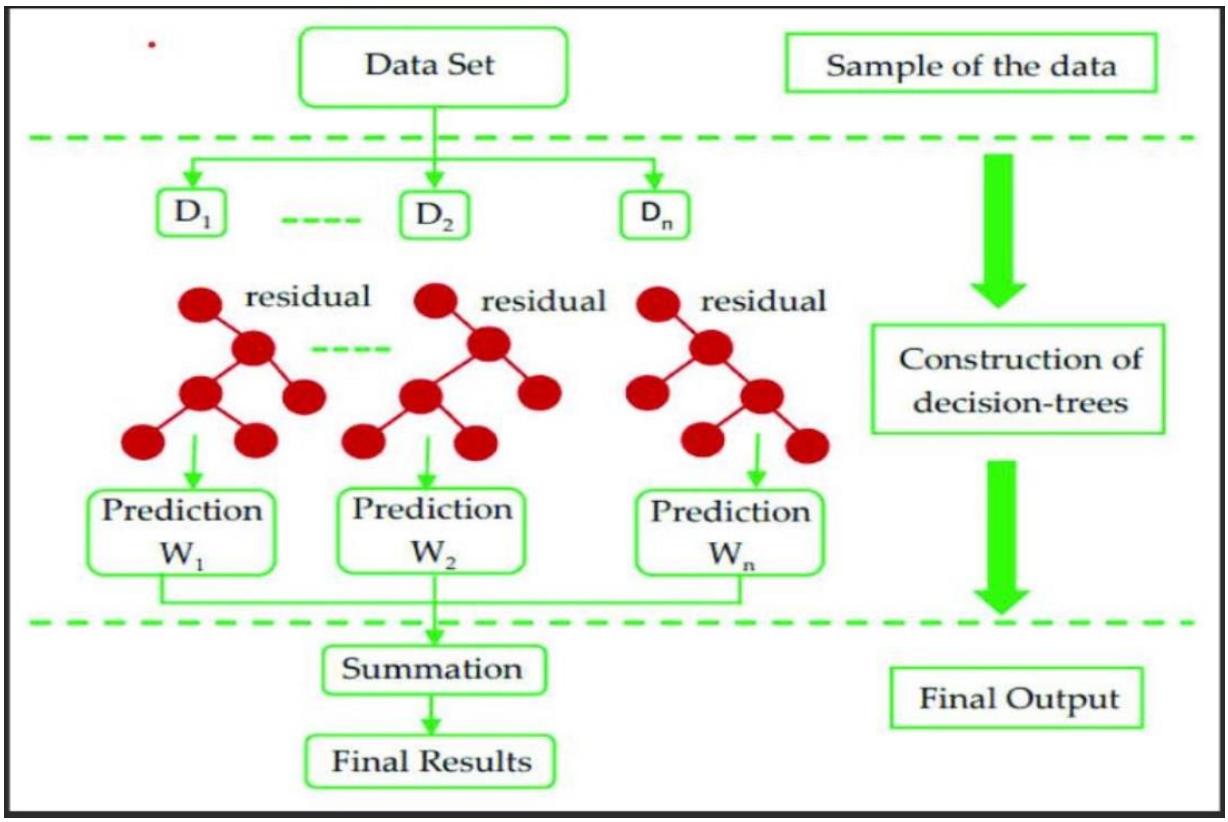


FIGURE 2 GADIENT BOOSTING MODEL

2.5.5. Decision Tree

Although decision trees are a supervised learning technique, they are primarily employed to solve classification problems. However, they can also be used to solve regression problems. This classifier is tree-structured, with internal nodes standing in for dataset attributes, branches for

decision rules, and leaf nodes for each outcome. The Decision Node and the Leaf Node are the two nodes that make up a decision tree.

While leaf nodes represent the result of decisions and do not have any more branches, decision nodes are used to make any kind of decision and have numerous branches. The characteristics of the provided data set are used to inform the decisions and testing. It is a graphical tool that shows all of the options for solving a problem or making a decision given certain parameters. It is named a decision tree because, like a tree, it begins with the root node and grows on subsequent branches to form a structure like a tree. (Rokach, n.d.)The Classification and Regression Tree algorithm, or CART algorithm, is used to construct trees. A decision tree only poses a query, and then divides the tree further into nodes based on the response (Yes/No)

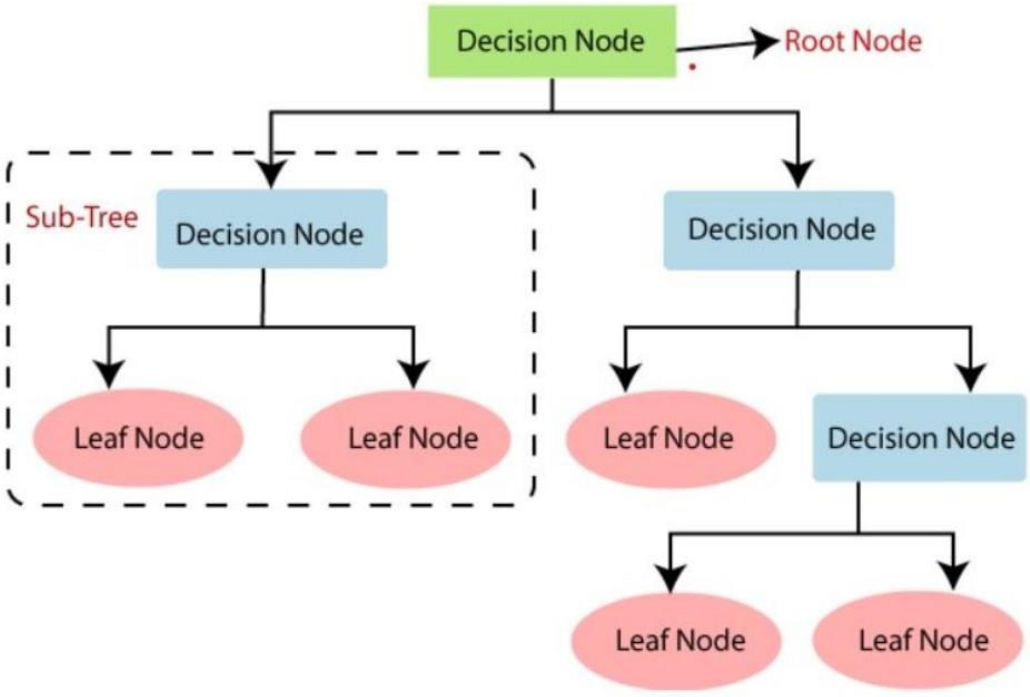


FIGURE 3 DECISION TREE MODEL

2.6 DEEP LEARNING APPROACHES

2.6.1. RNN

Recurrent neural networks, or RNNs, are specifically made to handle sequential data that is, data in which sequence is important. RNNs possess an internal memory, in contrast to conventional neural networks. They can "remember" details from earlier steps in the sequence thanks to this memory, which affects their predictions for the present phase. RNNs function in this way. At every time step, the RNN gets an input, such as a word in a sentence. Hidden State: The input is processed

in conjunction with the memory, which was the previous hidden state. Output: Using the current input and the hidden state as its bases, the RNN generates an output. Memory Updating: New information from the current input is reflected in the hidden state.

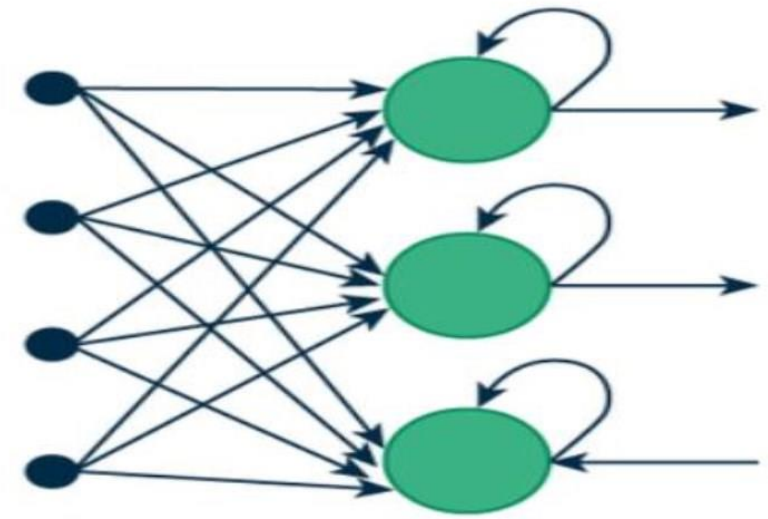


FIGURE 4 RNN ARCHITECTURE

The deep learning approach is a type of machine learning on an artificial neural network that can learn features of the text by the model itself. Deep learning approaches can handle context as compared with classical machine learning (Kuncoro et al., n.d.). For this study, we experimented with long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (Bi-LSTM).

2.6.2 LONG SHORT-TERM MEMORY (LSTM)

An example of a recurrent neural network with a memory for long-term data storage is the Long Short-Term Memory (LSTM). The LSTM's more complex structure makes it an excellent choice for handling the vanishing gradient problem. Long-term data and context are maintained when the LSTM is applied in any sequential activity. The vanishing gradient problem, which results in the loss of context, and a typical recurrent neural network are shown below. LSTM preserves context and information simultaneously. The basic difference between regular recurrent networks and LSTM is understandable.

Unlike other neural network architectures, the LSTM network is made up of layers of connected memory blocks as opposed to entangled neurons. The block's information flow, functionality, and

state are all controlled by gateways. Gateways can assess if a group of data should be kept around because it is significant. The greatest method for preserving long-range context is LSTM; unfortunately, it only captures forward context (one-directional). The following activities are performed by the LSTM using four memory block elements.

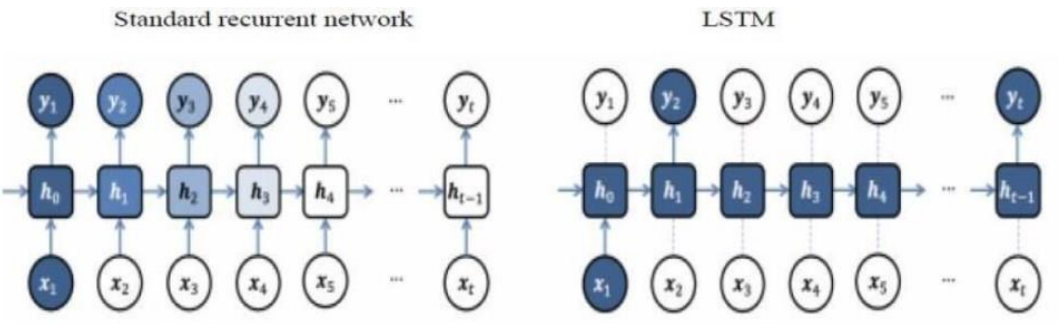


FIGURE 5 COMPARISON OF STANDARD RECURRENT NETWORK

- Input gate: is used to keep track of the data being entered into the memory block.
- Cell gate: is used to store long-term information.
- Forget gate: is utilized to decide which information should be kept and which should be eliminated.
- Output gate: based on the input and memory unit, utilized to decide what operation to do on the output.

This allows LSTM networks to selectively retain or discard information as it flows through the network, which enables them to learn long-term dependencies. The LSTM maintains a hidden state, which acts as the network's short-term memory. The hidden state is updated based on the input, the previous hidden state, and the memory cell's current state. Networks in LSTM architectures can be stacked to create deep architectures, enabling the learning of even more complex patterns and hierarchies in sequential data. Each LSTM layer in a stacked configuration captures different levels of abstraction and temporal dependencies within the input data.

LSTM architecture has a chain structure that contains four neural networks and different memory blocks called **cells**.LSTM networks are the most commonly used variation of Recurrent Neural Networks (RNNs). The critical component of the LSTM is the memory cell and the gates (including the forget gate but also the input gate), inner contents of the memory cell are modulated by the input gates and forget gates. Assuming that both of the segues are closed, the contents of the memory cell will remain unmodified between one time-step and the next gradient gating structure allowing information to be retained across many time steps, and consequently also allowing the group to flow across many time steps.

2.6.3. BIDIRECTIONAL LONG SHORT-TERM MEMORY

A subset of LSTM networks is Bidirectional Long Short-Term Memory (Bi-LSTM) networks. Bi-LSTMs are made up of two layers that are hidden from view. The input sequence is processed forward by the first hidden layer. However, the sequence is processed backward by the second hidden layer. Comparative Study between Traditional Machine Learning and Deep Learning Approaches for Text Classification. The output layer may access the past and future backgrounds of each point in the series to these hidden layers. The LSTM as well as its bidirectional variations proved to be incredibly beneficial. They may learn when and how to forget particular pieces of knowledge, as well as when and why not to use certain gateways in their architecture. The Hyperparameter variables govern the configuration of the network, including the number of hidden units and the training method (e.g., learning rate) of the network. Hyperparameters are set before training (i.e., before weight and bias optimization). Hyperparameters are important since they directly influence the behavior of the training model and greatly influence the model's performance. Here are the most popular hyperparameters:

Hidden layer the layers that lie between the input and output layers are known as hidden layers. Accuracy may be increased by several hidden units within a layer using regularization techniques. Reducing the number of units may result in fitting, and increasing the number of hidden layers will lead to overfitting. Thus, having an average amount of hidden layers is preferable.

Dropout is a regularization approach that enhances the generalizability of the model to be constructed and helps prevent overfitting by improving validation accuracy. Neural networks use calculations called activation functions to determine if a neuron can fire or not by calculating the weighted sum of input and biases. Depending on what it is used for, the activation function can be either linear or non-linear. Activation functions include Sigmoid, Rectified Linear Units (ReLU, Softplu), Softmax, Softsig, and Hyperbolic Tangent Function (Tanh). For binary classification models, the sigmoid activation function which was previously mentioned is frequently employed. The pace at which network parameters are updated is known as the learning rate. The learning process was hindered by a poor learning rate. The learning process moves more quickly at a higher learning rate. The total number of training epochs is the number of times the network shows all of the training data during the training process. The number of subsamples sent to the network after parameter adjustments happen is known as the batch size. Typically, the batch size is a power of two.

CHAPTER 3 METHODOLOGY

3.1. DATASET COLLECTION

Data related to idioms were collected from Amharic Idioms (A.Aklilu and D.Worku.1992. A, 1992), Debebe H/giorgis, (Haddis Alemayehu, 1996), and literal expressions from different Amharic documents. It is supervised learning that manually annotates the data set. All the collected data would be cleaned, removed, stop words, numbers, and normalized characters, and finally prepared for model training, evaluation, and deployment.

The models would be trained using these datasets. To test the model, arbitrary phrases from different Amharic sources would be selected, a combination of literals and idioms from the Amharic Idiom book. A compilation of literary works from various Amharic texts and idiomatic phrases from books containing Amharic idioms was put together. We can understand the meaning and characteristics of Amharic idioms despite their diverse origin and production in terms of people's bodies, lifestyles, or other issues. This study aims to identify phrase-level idioms, or idioms that are composed of two words. The character of the expression makes idiom identification a difficult and confusing procedure. This study contributed to the demonstration of the feasibility of automating idioms through the use of machine-learning techniques. An Amharic idioms book, along with other Amharic texts, was used to gather 5,500 idiomatic and literal expressions for training and testing in the proposed model. The expressions are divided into two categories: literal (2750) and idiomatic (2750).

3.2. DATASET ANNOTATION

Data annotation is the process of labeling data to help machine learning models understand and classify information. It's a crucial step in training AI models and was used in supervised learning and semi-supervised machine learning models. After the dataset had been gathered, an expert in language annotation classified certain expressions as idioms or non-idioms. The classification of the dataset into idiomatic and literal expressions is carried out once the labeling is complete.(Seid Muhie Yimam, 2021) .The annotation process of a given phrase is shown in the picture.

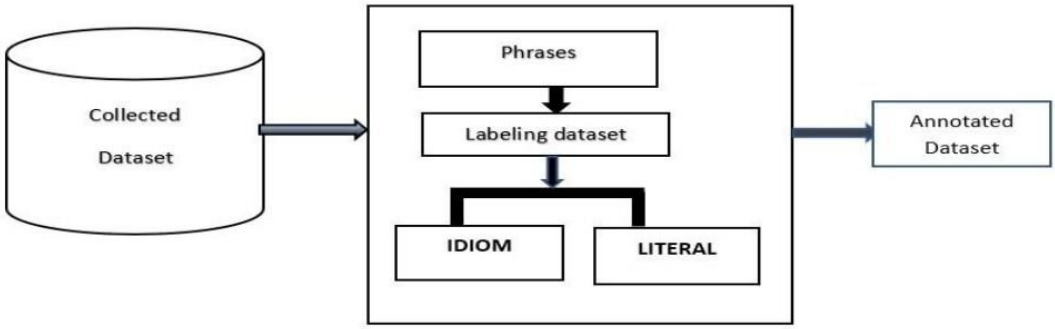


FIGURE 6 ANNOTATION PROCESS

3.2.1. DEPENDENT VARIABLE

In the context of Amharic language idiom identification, the dependent variable typically refers to the outcome or the variable that is being measured or predicted. In this case, the dependent variable would be: Idiomatic Expression Classification: This variable indicates whether a given expression is classified as an idiom or a non-idiom (literal expression). (Kelemework, 2013)

The goal of the model would be to correctly identify and classify these expressions based on the features extracted from the dataset. There are different kinds of features in idiomatic expression classification.

Lexical Features: Consider the implications and meanings of the phrases used in the sentence. Lexical features are the words and how often or how they are used in a text. Lexical characteristics are widely employed as input variables in machine learning models. These attributes are converted into numerical representations that models may process using methods like bag-of-words or word embeddings (e.g., Word2Vec, GloVe). Sequences of lexical characteristics can be used by deep learning models, especially transformers and recurrent neural networks (RNNs), to acquire contextual meanings. Syntactic Structure: Determine the idiom's parts of speech and sentence construction as well as its grammatical structure. The syntactic structure involves the arrangement of words and phrases to create well-formed sentences, often represented through parse trees or dependency graphs offering details on the grammatical connections between words, syntactic features can improve the performance of machine learning models. Models that make use of this structure include parse-based neural networks. Tasks like sentiment analysis, translation, and text classification can all be made better by including syntactic information.

Morphological Variations: Take into account changes to word forms, including inflections, and

suffixes, and understanding meaning can be greatly aided by morphological characteristics, particularly in languages with complex morphology. These properties can be included in machine learning models using methods like lemmatization and stemming. Deep learning methods are better able to handle uncommon or out-of-vocabulary words because they can catch morphological variants more effectively especially when using character-level embeddings.

3.2.2. INDEPENDENT VARIABLES

Identifying independent variables for Amharic idioms requires considering linguistic, contextual, and cultural factors. These variables are chosen or altered to forecast the value of a dependent variable. They are the model's input. Textual Features in NLP: Using techniques like TF-IDF, word or phrase embeddings, POS tags, n-grams, etc., text data can be converted into numerical representations; these are independent variables.

3.2.3. CONTROL VARIABLES

Control variables are factoring that researchers keep constant to isolate the effects of independent variables in a study. When identifying Amharic idioms, controlling for certain variables can help ensure that the results are not influenced by external factors. Control variables are those that are kept constant or taken into consideration during research to determine how the independent factors affect the dependent variable. They assist in declining confusing biases.

3.2.4. ANNOTATION QUALITY

Annotation quality refers to the accuracy, consistency, and reliability of the labeling or categorization of data, which is crucial for tasks like Amharic idiom identification. High-quality annotations are essential for developing robust machine-learning models and linguistic analyses. Here are key aspects of annotation quality specific to Amharic idiom identification: Deep learning and annotation quality go hand in hand because well-written annotations greatly influence the efficiency and performance of deep learning models. The salient features of this relationship are as follows: Preprocessing Techniques: The choices you make in tokenization, stemming, stop word removal, etc.

3.3. TEXT PREPROCESSING

Text preprocessing is a crucial step in machine learning, particularly in natural language processing (NLP). It involves preparing and cleaning text data to facilitate better analysis and model performance. Amharic phrase-level data refers to transforming raw text into a more consistent and standardized format, making it easier to analyze and process. This is particularly important for natural language processing tasks, including idiom identification. Key aspects of text normalization for Amharic might include:

1. Cleaning: Remove any irrelevant data interfering with idiom identification, such as HTML tags, URLs, numbers, or non-Amharic text.
2. Stop Word Removal: Filtering out common stop words (e.g., "nebere" meaning "was" “new” meaning “is”) that may not add significant meaning to the analysis. (Sileshi Girmaw Miretie, n.d.)
3. Stemming and Lemmatization: Reducing words to their root forms. For example, different forms of a verb may be converted to a single base form.
4. Whitespace Removal: Eliminating unnecessary spaces or tabs to ensure clean text input. Whitespace removal refers to the process of eliminating unnecessary whitespace characters from text data. Whitespace characters include spaces, tabs, and newline characters.
5. Handling Special Characters: Converting or removing special characters that may not be relevant or may cause confusion.
6. Tokenization for Amharic phrase-level data is the process of breaking down a continuous text into smaller units, or "tokens," which can be words, phrases, or sub-words. This is a crucial step in natural language processing (NLP) as it helps analyze and understand the text's structure and meaning. Amharic tokenization can be challenging due to the script’s complexity and the language’s agglutinative nature.
7. Word Tokenization: This involves splitting the text into individual words. Given that Amharic uses a script that can connect characters, special care must be taken to identify word boundaries accurately. For example, the phrase "እሱ መጣ " (he is coming) would be tokenized into ["እሱ", "መጣ"].
8. Part-of-Speech Tagging: Annotate each token with its corresponding part of speech, which can be crucial for recognizing idiomatic structures.
9. Text Normalization: Convert all text to a uniform format, such as using the same script or character set for Amharic, and ensuring consistent use of diacritics and punctuation. In Amharic NLP research, character normalization is conceptualized in two ways.

Most NLP research involves trend normalization as a preprocessing step; however, others claim that Amharic symbols and alphabets have different meanings and should not be subjected to normalization. The opposing party contends that characters sharing the same phoneme or sound have no unique meanings and that all phonemes should be represented by a single representative alphabet, thus normalizing the symbols. Normalization is required to eliminate redundant words that have the same meaning because a single Amharic character can be expressed in multiple ways. In light of this fact, we have applied normalization to eliminate the repetition of terms in distinct representations. Some characters that are similar in sound in Amharic are (ሀ፣ሃ፣ሐ፣ኀ፣ኃ፣ኻ), normalized as "ሀ", (ዐ፣ዓ፣አ፣ኣ), normalized as "አ", (ጸ፣ፀ), normalized as "ፀ", and (ሰ፣ሠ) normalized as "ሰ". To achieve this, all variations of such characters with the same sound merge into a single form using the normalizing technique.

In the context you've provided, an identifier refers to a component of a machine-learning model that recognizes and categorizes idiomatic expressions. The model utilizes a supervised machine learning algorithm, meaning it learns from labeled data where the idiomatic expressions are already identified. During training, the algorithm analyzes features extracted from the data these could include linguistic characteristics or patterns associated with idiomatic expressions. The vector representation refers to a mathematical way of representing these expressions in a format that the algorithm can process. Each idiomatic expression is transformed into a numerical vector, capturing its properties and relationships with other expressions. Ultimately, the identifier in this model serves to distinguish idiomatic expressions from non-idiomatic ones based on the learned patterns from the training data, enabling it to recognize similar expressions in new, unseen data.

3.4. WORD REPRESENTATION

This stage involved identifying text feature extraction methods that are effective for idiomatic expression identification. After the text dataset has been preprocessed, it needs to be translated into vector format so that the machine learning or deep learning model can interpret it. To properly grasp and process language, deep learning models require the conversion of text data into a format that is compatible with them. This is where the Keras Embedding layer comes into play. The layer improves the performance of the model by capturing the semantic meaning of words through the process of learning embeddings.

We used TFIDF for the machine learning technique. However, text dataset embedding can be computed using the Keras embedding layer for neural networks. Each word in the input data must have a distinct number in an integer encoding. The embedding layer learns an embedding for each sentence in the training dataset, starting with random weights.

The techniques used to translate words or phrases into numerical vectors that describe their semantic features, relationships, and meanings are referred to as word representation. These representations facilitate the efficient processing and comprehension of human language by machines. There are various widely used methods for representing words:

3.5. FEATURE EXTRACTION

3.5.1. TF-IDF

Term Frequency–Inverse Document Frequency is referred to as TF-IDF. It is a statistical metric for assessing a word's significance in a document about a group of documents (corpus). Words and their TFIDF scores are represented by each dimension in a vector representation of a document that is frequently made using TF-IDF. It is employed for keyword identification and feature extraction; we use this TF-IDF for the machine learning model.

3.5.2. COUNT VECTORIZER

Natural language processing (NLP) uses a count vectorizer to transform text data into a numerical representation that machine learning algorithms can comprehend. In essence, it generates a vector representation of a document by counting the instances of each word in the document. Lexical content. To build a vocabulary of distinct terms, the Count Vectorizer first examines the complete corpus of text, which is a collection of documents. The vector representation is built upon this vocabulary. The Count Vectorizer keeps track of the number of times each vocabulary word appears in a given document.

3.5.3. ONE HOT ENCODER

In machine learning, a One-Hot Encoder is a method for transforming category data into a numerical representation that algorithms may use. For every category, it generates a binary vector with one element set to 1 and all others set to 0. (J. Peng A., 2016). The categorical feature's unique categories are first all identified by the One-Hot Encoder. A binary vector with the same length as the total number of unique categories is made for each category. Except for the element belonging to that category, which is set to 1, the vector is entirely composed of zeros.

3.5.4. LABEL ENCODING

In machine learning, label encoding is a technique that transforms category data into numerical data. It gives every feature category a distinct integer. Label encoding just substitutes a

numerical value for the category, as opposed to One-Hot Encoding, which generates a binary vector for every category.

3.5.5. PADDING

In machine learning, particularly deep learning, padding is the process of appending extra data (often zeros) to the start or finish of a sequence to guarantee that every sequence in a dataset has the same length. This is essential when working with models such as recurrent neural networks (RNNs) that require inputs to be fixed in length.

3.6. DEVELOPMENT TOOLS

We conducted an experiment whereby the Python programming language was utilized for the preprocessing, model training, and model testing tasks. Python is a general-purpose, high-level programming language that is also relatively easy to use code editors with different Python versions. packages are used with the Anaconda Jupiter notebook Python framework. Backend computations are performed using Keras deep learning frameworks. The constructed model is tabulated and analyzed using several Python tools such as NumPy, Panda s, and matplotlib.

Python is an object-oriented, interpreted, interactive, high-level programming language. It is meant to be an extremely understandable language, Compared to other languages. its qualities as a robust, general-purpose, and user-friendly programming language, Python was our choice for the implementation. Keras is an application programming interface (API) for Python-based high-level neural networks. This open-source neural network library aims to facilitate the rapid creation of models based on deep neural networks. We constructed Keras in this study on top of TensorFlow. Keras is an intuitive deep-learning library that is easy to use, modular, and extensible. Keras for deep learning enables fast prototyping and faultless execution on both CPU and GPU. We have taken into consideration the previously listed attributes and have made use of the Keras deep learning package. The most robust and easily navigable machine-learning library is called Scikit-learn, or Sklearn. It employs a suite of effective tools for statistical modeling and machine learning, including dimensionality reduction, clustering, regression, and classification, which are provided via the Python consistency interface. Primarily written in Python, this package relies on Numpy, Scipy, and Matplotlib.

The Python library known as Numpy, which stands for "Numerical Python" or "Numeric Python," enables speedy mathematical calculations on arrays and matrices. Therefore, in this study, we describe our text dataset numerically using arrays using the NumPy Python module. Pandas is comparable to NumPy, one of the most widely used Python packages. It provides an in-memory 2D table object called a Data frame, which allows it to provide high-performance assemblies and

easy-to-use file analysis capabilities. It offers multi-dimensional array objects, similar to spreadsheets, with labels for each column and row. This package is what we use to load and manage our dataset for our study. Matplotlib is a Python programming language visual charting toolkit. It is used to plot multiple findings and display different graphs during the model training process.

3.7. DESIGNING AMHARIC IDIOMS IDENTIFICATION MODEL

For the proposed work, we used an experimental research design methodology that allows us to adjust various Hyperparameters and experimentation setups to examine their impact on the suggested model. (Muzny & Zettlemoyer, n.d.)Several experimental configurations were used with this research approach, and their impact on the suggested study project was assessed. for us to assess and comprehend how different factors affect the study methods that are intended to produce a solution. In this research procedure, certain factors are changed to see how they impact other variables. The variables in consideration are datasets, experimental conditions, and model hyperparameters. In this thesis, these factors are varied to investigate their effects and find the best-fit configuration. Overall, the method of experimental study has helped us measure.

The suggested work is constructed using a deep-learning methodology. The steps of the proposed work were the acquisition of the dataset, annotation of the dataset, preprocessing of the dataset, model building, classification, identification of idiom, and evaluation. Gathering phrases that have and do not have Amharic idioms is the primary task of this effort. After the datasets are gathered, domain experts annotate them so the model may be tested and trained. Following the annotation subtask's completion, the dataset is preprocessed through the steps for text preprocessing, which includes stop word removal, normalization, and punctuation mark and number removal. Converting gathered data into a format that the model can be trained and tested. Since machines can’t understand text data directly as we humans do, it is a must to apply different word representation techniques to change the data into a numeric vector. After representation, the model is trained and tested the performance. We have experimented with LSTM, Bi-LSTM, SVM, ensemble, and Gradient boosting.

3.8. MODEL ARCHITECTURE

The four components of a proposed model architecture are preprocessing, word representation, model development, and model testing. First, the preprocessing stage is finished, and the representation dataset is converted into numerical vectors using the word embedding technique, arranging it for the suggested model construction. Next, a model for the categorization and identification of idiomatic expressions was built and trained. To assess how well the LSTM and

Bi-LSTM algorithms fared in their predictions for the recommended task, this section required training and testing data. In the end, the created model was evaluated by applying machine learning performance measure criteria. The suggested model architecture is shown in Figure 7.

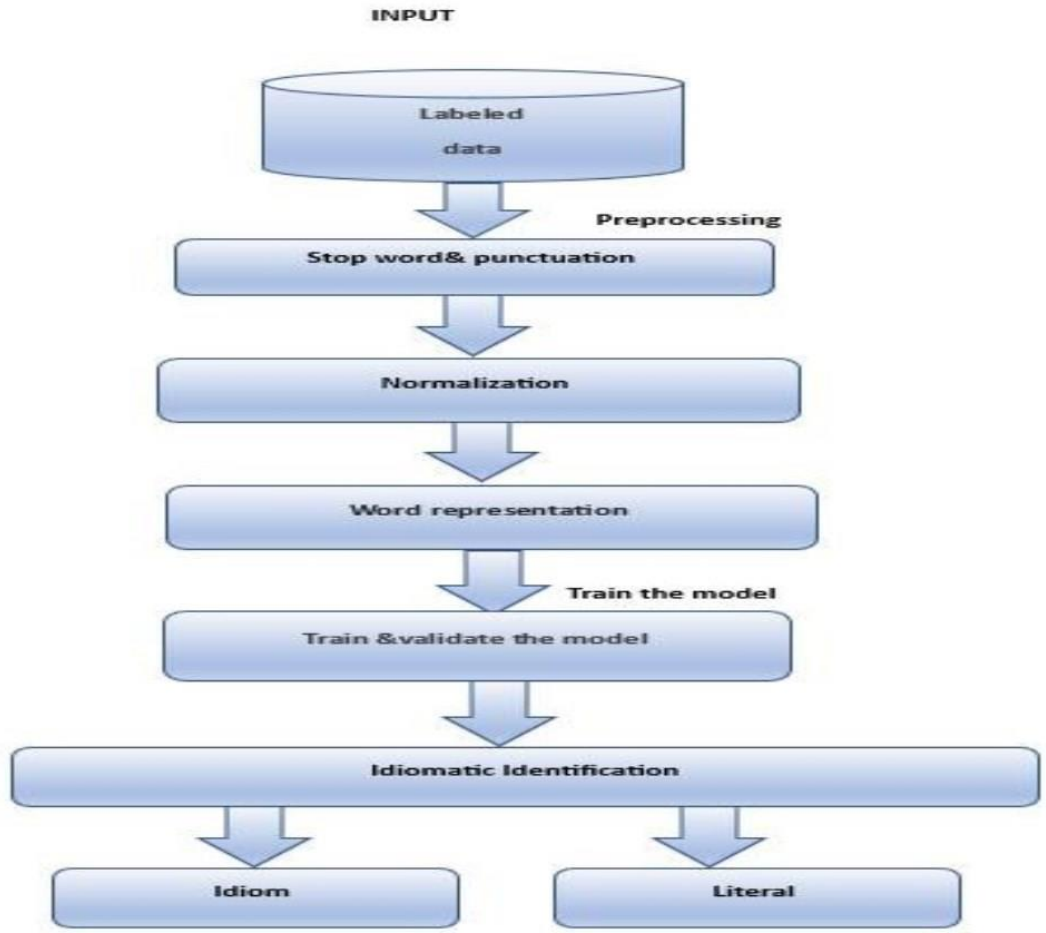


FIGURE 7 MODEL ARCHITECTURE

3.9. MODEL DEVELOPMENT

The actual task in this step is identification based on the classification of idiomatic expressions. To begin the classification process, the input sentence is divided into classes for idioms and non-idioms. If the input statement is classified as an idiom, the process of identifying and designating idiomatic expression terms will follow. For our idiomatic expression identification model, three approaches are available: dictionary-based, traditional machine learning, and deep learning approaches. From those methods, we have experimented to create the model utilizing machine

learning (SVM, Gradient boosting, and Ensemble) and deep learning (LSTM, and Bi-LSTM) algorithms.

The Bi-LSTM modeling and long-term memory feature of LSTM enable it to handle context well. The fact that words' forward and backward contexts are not maintained is one of LSTM's shortcomings. This is due to the forward-only nature of the neural network used in LSTM. LSTM and Bi-LSTM (Kamath C. N., 2018). Compared to LSTM, Bi-LSTM performs better in comprehending word context. Bi-LSTM has two distinct hidden layers, which explains this. The first hidden layer is where the forward input sequence is processed. Processing the sequence backward is how the second hidden layer works. Bi-LSTM is better at accurately capturing word meaning because of its two hidden levels. Bi-LSTMs provide a more comprehensive understanding of sequential data by considering both past and future information, making them a powerful tool for many applications in machine learning, particularly in fields that rely heavily on the context of sequences.

3.10. MODEL PERFORMANCE EVALUATION

The performance evaluation of the generated model is the final stage of this inquiry job. Precision, Recall, F-score, and accuracy are the assessment measures used in this work to assess the constructed model. Due to their ability to distinguish between correctly categorized and incorrectly classified data within the dataset, those assessment metrics are effective for measuring textual data. The percentage of idiomatic and non-idiomatic expressions that are accurately anticipated is called recall. The percentage of expected idiomatic and non-idiomatic utterances is measured by precision. Recall and precision are balanced performance measures that are analyzed using the F-score measure. Another way to assess the overall efficacy of the suggested model is through accuracy. Categorization data, which are often presented in a Confusion matrix format, can be used to construct the previously described performance indicator.

3.11. CONFUSION MATRIX

A confusion matrix is a table that displays the effectiveness of a categorizing technique. It is also used to illustrate and summarize the performance of a grouping process. Four outputs are used in the confusion matrix to indicate counts from actual and expected values: The number of idiomatic expressions that were successfully detected with this technique. True Positive (TP) is the correct prediction of idiomatic outcomes. The True Negative (TN) metric indicates the proportion of non-idiomatic phrases that are accurately classified as literal. False Positive (FP): displays the number of non-idiomatic expressions that are incorrectly identified as idiomatic.

False Negative (FN): indicates the proportion of idiomatic expressions that are incorrectly labeled as non-idiomatic. The following table shows the predicted value of the confusion matrix.

	Actual Value		
Predicted Value		Idiom	Literal
	Idiom	TP	FP
	Literal	FN	TN

TABLE 2 CONFUSION MATRIX

Precision is the proportion of the number of positive examples classified over all the examples classified.

$$\text{Precision (P)} = \frac{TP}{TP+FP}$$

Recall is the proportion of the number of positive examples classified over all the positive examples.

$$\text{Recall (R)} = \frac{TP}{TP + FN}$$

F1- measure is the normalized value of the two-measurement metrics precision and recall performance measures.

$$\text{F1-measure } F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The simplest evaluation metric is accuracy so the overall effectiveness of the algorithm is calculated by dividing the correct labeling against all detections.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

Accuracy gives a general idea of the model's performance. Recall focuses on the model's ability to identify all relevant instances. Precision ensures that the positive classifications are indeed accurate.

These metrics are crucial for understanding the effectiveness of classification models, especially in scenarios where class distribution is imbalanced.

3.12. Summary

This chapter comprehensively covered various aspects of our study methodology, emphasizing the systematic approaches we employed throughout our research. We began by detailing how we gathered a diverse range of texts from various Amharic books. Each of these texts was carefully annotated or labeled to enhance the quality of our dataset. In addition to our data collection efforts, we implemented rigorous text preparation methods to ensure the data was suitable for analysis. This involved developing a preprocessing technique aimed at eliminating irrelevant components such as stop words and punctuation. Furthermore, we applied normalization techniques to standardize the text, ensuring consistency across our dataset.

To facilitate our investigation, we utilized an environment specifically designed for writing Python programs, allowing us to leverage its robust libraries and frameworks. Our experimental methodology was meticulously planned to ensure that we could effectively test our hypotheses and validate our findings. We conducted extensive tests on various word representation techniques, including dense embedding, TF-IDF, and Keras embedding, to determine the most effective approach for capturing the idiomatic expression of the Amharic language.

In our exploration of Amharic idiomatic identification, we employed several machine learning algorithms, such as Support Vector Machines (SVM) and gradient boosting techniques. Additionally, we integrated ensemble methods to enhance our model's performance. we also utilized deep learning methods, specifically Long Short-Term Memory (LSTM) networks, and Bidirectional LSTM (Bi-LSTM) networks, to capture complex patterns and relationships within the data. Through this complex approach, we aimed to provide a thorough and insightful investigation into the workings of Amharic idiomatic expressions, leveraging a combination of traditional and pioneering methodologies.

CHAPTER 4 RESULT AND DISCUSSION

4.1. EXPERIMENTATION

This chapter includes the dataset utilized, the evaluated algorithm, the results of the tests conducted, and the performance indicators for the proposed model. We have collected 5,500 Amharic phrases for classification-based identification, both idiomatic and non-idiomatic, to conduct experiments. To preserve data balance, 2,750 phrases have idioms in these datasets whereas the remaining 2,750 phrases do not. We used SVM, Gradient Boosting from machine learning LSTM, and Bi-LSTM from deep learning to develop the model. employing a support vector machine model for the ensemble with Bi-LSTM and Gradient Boosting.

4.2. EXPERIMENTATION SETUPS

4.2.1. DATASET DESCRIPTION AND DISTRIBUTION

To identify idiomatic expressions during the model-building process, we used 5,500 Amharic phrases. The distribution of the datasets into training and testing groups is summarized in the table. Phrases with a minimum length of five words and a maximum length of ten are included in the dataset. Since machine learning and deep learning require input that is the same size, we used a technique called padding to ensure that all of the input sequence data had the same length as the input points.

Idiomatic Identification Dataset			
	Training	Testing	Total
Idiomatic	2,200	550	2,750
Literal	2,200	550	2,750
Total	4,400	1,100	5,500

TABLE 3 DATASET SPLITTING

The dataset is split into training and testing which is presented in (Table 4) employing a classifier with an 80,20-splitting ratio If the training set is too large, the model may overfit the data, capturing noise rather than the underlying patterns. This can result in poor performance on new, unseen data.

Conversely, if the testing set is too small, the model may underfit, failing to adequately represent the complexity of the data, which can also lead to sub-optimal performance. Essentially, selecting the data ratio during dataset partitioning is a trade-off between giving the model access to enough data for learning and guaranteeing a reliable assessment of its output. To find the best split for a particular dataset and machine learning task, it is imperative to test several ratios and evaluate the model's performance. We find that our model performs well at an 80:20 splitting ratio. We selected an 80:20 ratio, meaning that 80% of the data is utilized for training and 20% is used for testing. In our experiments, we employed Bi-LSTM, and the accuracy was 98.18%. I experimented with different splitting methods, but I prefer the 80:20 ratio because it is suitable for my dataset.

4.2.2. ENVIRONMENT AND HYPERPARAMETER SETUPS

We set up environmental and Hyperparameter setups for our experimentation. We used an HP laptop with an Intel(R) Core (TM) i7-1255U CPU@ 1.70 GHz processor and 16 GB RAM for training and testing. To evaluate the selected algorithms LSTM, and Bi-LSTM for idiomatic expression identification. Keras embedding and dense embedding are used as word representation for the deep learning approach whereas, TFIDF word representation is used for the machine learning approach to represent the terms in vector form. We have used different kinds of hyperparameters. Hyperparameter setups refer to the configuration of the adjustable parameters in a machine learning model or algorithm that are not learned during the training process. These hyperparameters are set before the training begins and have a significant impact on the model's performance.

No	Hyperparameter	Experiment 1	Experiment2
1	Batch Size	128	64
2	Learning Rate	0.01	0.0001
3	Dropout	0.5	0.2
4	Activation Function	Tanh	Softmax
5	Epoch	10	20
6	Optimizer	Nadam	Adam
7	Sequence Length	23	23
8	Embedding Dimension	128	200

TABLE 4 HYPERPARAMETER SETUP

LSTM Layer Count: The model's total number of LSTM layers the representation capacity of the model may be enhanced by adding more layers, although doing so raises the possibility of overfitting. LSTM Unit Size: This hyperparameter controls how many units, or neurons, are present in each LSTM layer. It also affects how complex the model is and how much data it can extract from the input sequence. Activation functions play a crucial role in neural networks and deep learning models. Softmax, Tanh, or Sigmoid are the most common activation functions utilized in LSTM cells.

Dropout Rate: To avoid overfitting, the LSTM layers are subjected to a dropout rate. Batch Size: The number of samples utilized in each training cycle; higher batch sizes increase the rate at which the model converges, but they also increase memory requirements. With a few features added specifically for this work, the BILSTM model shares the same hyperparameters as the LSTM model. The learning rate determines the size of the steps taken toward the minimum of the loss function during optimization. A higher learning rate results in larger updates, while a lower learning rate results in smaller updates. The step size at which the model's parameters are updated during training by an optimization algorithm, such as Adam or SGD.An epoch signifies that the model has seen and learned from all the samples in the training dataset once. This is a fundamental unit of training time. The number of epochs is the number of times the model is trained using the whole training dataset.

Bi-LSTM analyzes the input sequence both forward and backward, enabling the model to capture information from both contexts. This is the primary distinction between Bi-LSTM and LSTM. This can be especially helpful for jobs like natural language processing that need a thorough comprehension of the input sequence. After attempting multiple successful configurations, the aforementioned hyperparameters were selected for testing (Experiments 1, and 2). The previously mentioned Hyperparameter configurations serve as the foundation for the model that we are showcasing. The Hyperparameter configuration with 64 batch sizes, 0.2 dropouts 20 epochs is effective for training and testing. Through testing, we discovered that we can improve training and prediction/testing performance by using the aforementioned Hyperparameter value. While too little of a batch size necessitates a lengthy training time, which adds noise to our model, large batches lead the model to have poor generalizing ability, which degrades the model performance. As a result, the intermediate batch size is 32. A high or low learning rate value mustn't cause the loss function to drop to the optimal range. As a result, the learning rate of 0.0001 works well for our task since it increases accuracy while decreasing loss.

It is essential to apply a dropout layer with a preset value to avoid the model becoming overfitting. When the lowest dropout is utilized, underfitting occurs; when the maximum dropout is used, overfitting occurs. Therefore, we decided on a 0.2 dropout value, which improved accuracy while

lowering loss, to address the overfitting problem in our work. The model overlearns when the epoch value is high, and underearns when it is low, which degrades the model's performance.

4.3. HYPERPARAMETER SETUPS FOR MACHINE LEARNING

4.3.1. SVM (SUPPORT VECTOR MACHINE)

We've utilized the Grid Search hyperparameter, varying the values of "C," "gamma," and "kernels." The regularization parameter manages the trade-off between decreasing the complexity of the decision boundary and obtaining a low error on the training set. A more complex model is produced by larger values of C. Alpha Kernel Coefficient It indicates the extent to which a single training example has an impact. A large similarity radius, indicated by a low gamma value, results in a smoother decision boundary. Kernel: Indicates the kind of kernel that will be applied to the algorithm. It can be "linear," "poly" (polynomial), or "rbf" (radial basis function). All things considered, adjusting these hyperparameters using Grid Search makes it possible to identify the best combination for maximizing the model's performance on the specified dataset. It is beneficial.

4.3.2. ENSEMBLE SVM WITH GRADIENT BOOSTING

In an ensemble model, the combination of Gradient Boosting and Support Vector Machines (SVM) methods can offer many benefits. Enhanced Capabilities in Forecasting Generally, ensemble models perform better than individual models because they take advantage of the advantages of many techniques. When managing nonlinear relationships, high-dimensional data, and noisy or unbalanced datasets, SVM and Gradient Boosting differ significantly. When these two algorithms are used together, rather than separately, the predicted accuracy can be improved. Additionally, ensemble models demonstrate increased robustness, exhibiting less sensitivity to the strengths and weaknesses of individual algorithms.

The other algorithms in the ensemble can make up for a single algorithm's poor performance on a given task, strengthening the overall model's resistance to the shortcomings of individual models. Managing the Variations in Data Characteristics: Different methodologies are used by SVM and Gradient Boosting to handle different types of data. While Gradient Boosting works well in handling noisy data and capturing complex feature interactions, Support Vector Machines (SVM) are better known for their capacity to handle high-dimensional data and nonlinear relationships. The ensemble model may handle a greater variety of data kinds and problem complexities more effectively by combining various approaches. Flexibility in Hyperparameter Tuning: By combining SVM with

Gradient Boosting, hyperparameter tuning is made more flexible. The hyperparameters of each algorithm can be adjusted separately, allowing the ensemble model to take advantage of the various strengths of the individual models through appropriate hyperparameter tuning. Enhanced Capability to Explain When it comes to explaining things, ensemble models can be more helpful than individual black-box models. Understanding the model's decision-making process can be aided by gaining insights into the underlying drivers of the predictions through an examination of the contributions made by each model within the ensemble. It's crucial to remember that the particular benefits of the ensemble model will vary depending on the issue at hand, the properties of the data, and the specifics of how the ensemble strategy is implemented. To find the ideal ensemble configuration, a thorough assessment and comparison with individual models are required.

4.4. EXPERIMENTATION RESULT OF MACHINE LEARNING

The training and testing capabilities of the SVM and gradient boosting models for the categorization of idiomatic expressions have been discussed in this section. The performance outcomes of the chosen models are shown here, as we have already entered the Hyperparameter configurations. Experimentation Result of SVM model for identification of idiomatic expression. The performance result of the SVM model according to the different hyperparameters is presented as follows.

1. SVM

Experiments		Precision	Recall	F1score	Accuracy
Exp. 1	C[0.1]ga[0.01]kernel[poly]	97%	96%	97%	97 %
Exp.2	C[1]ga[0.001]kernel[rbf]	97%	96%	97%	97 %
Exp. 3	C[10]]ga[0.1]kernel[linear]	97.89 %	97.82%	98%	98 %

TABLE 5 SVM MODEL PERFORMANCE

As described in Table 6 support vector machine model achieved good performance in experiment three in terms of accuracy including precision, recall, and f1-score measurement metrics. However, showed poor performance in experiments 1 and 2 as we compared experiment 3. We understand from this experiment when the hyperparameter varies the value we got is different value. it works best in C,10 gamma 0.1 kernel linear.

FIGURE 8 PERFORMANCE REPRESENTATION OF SVM

2. Gradient Boosting

Experiments		Precision	Recall	F1score	Accuracy
Exp. 1	LR (0.001), MD (3), n-s (50)	78%	63%	57%	63%
Exp. 2	LR (0.01), MD (5), n-s (100)	78%	64%	58%	64%
Exp.3	LR (0.1), MD (7), n-s (200)	79%	65.5%	59%	65.55%

TABLE 6 GRADIENT BOOSTING PERFORMANCE MEASURE

Gradient Boosting has several important hyperparameters that can be tuned to optimize model performance.LR (value), MD (value), n-s(value): These likely represent the hyperparameters used in each experiment. Learning rate, or LR, is a crucial variable in optimization algorithms that regulates the step size when training models. By taking smaller steps, a lower learning rate (such as 0.001) can assist in preventing overshooting the ideal solution.MD (Model Depth): This probably relates to the model's depth, such as a neural network's layer count. Although a deeper model may be more prone to overfitting, it may also be able to learn more complex patterns. A model's training set of data points

is denoted by the symbol n-s (Number of Samples). Although more data can lengthen training times, it also often improves generalization.

Comparing Experiment 1, Experiment 2, and Experiment 3, the results show that Experiment 3 has the highest test and validation accuracy and training accuracy (65.55%). This suggests that even if the model is learning more complicated patterns, further raising the learning rate, model depth, and number of samples can improve generalization.

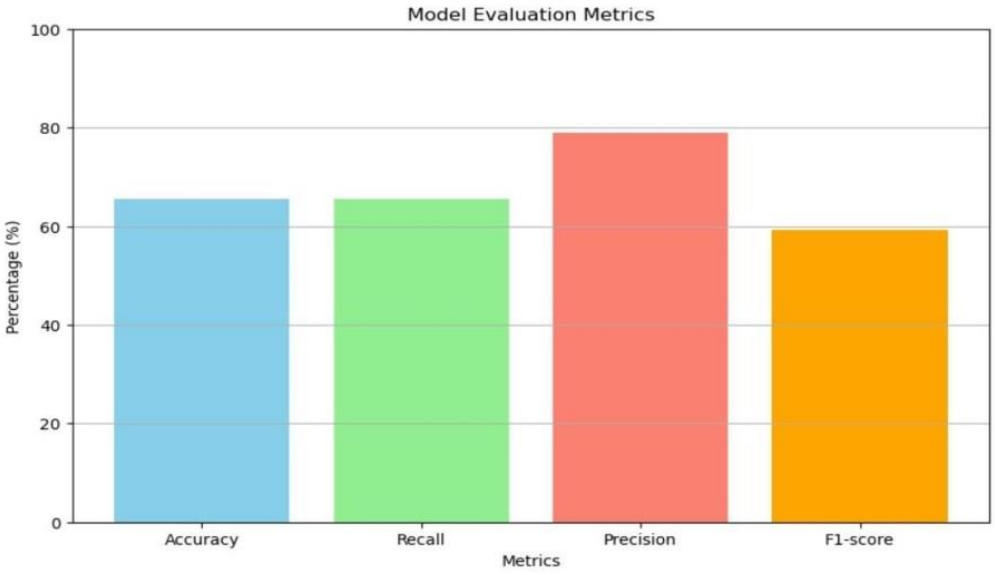


FIGURE 9 PERFORMANCE REPRESENTATION OF GRADIENT BOOSTING

3. Ensemble SVM with gradient boosting

Experiments		Precision	Recall	F1score	Accuracy
Exp.1	LR (0.01), MD (5), n-s (50)	97%	97%	97%	97%
Exp. 2	LR (0.1), MD (3), n-s (100)	98%	98%	98%	98%

TABLE 7 ENSEMBLE SVM WITH GRADIENT

As can be seen from the Table, the ensemble model outperformed the individual algorithms. This is because ensemble learning, an effective machine learning method, mixes several models to enhance prediction performance. By combining the advantages of multiple algorithms, this method produces forecasts that are more accurate than those of any one model working alone. The optimal

hyperparameters are n-estimators (n-s (100)), max-depth (MD (3)), and learning rate (LR (0.1)). These are the best hyperparameters for this model.

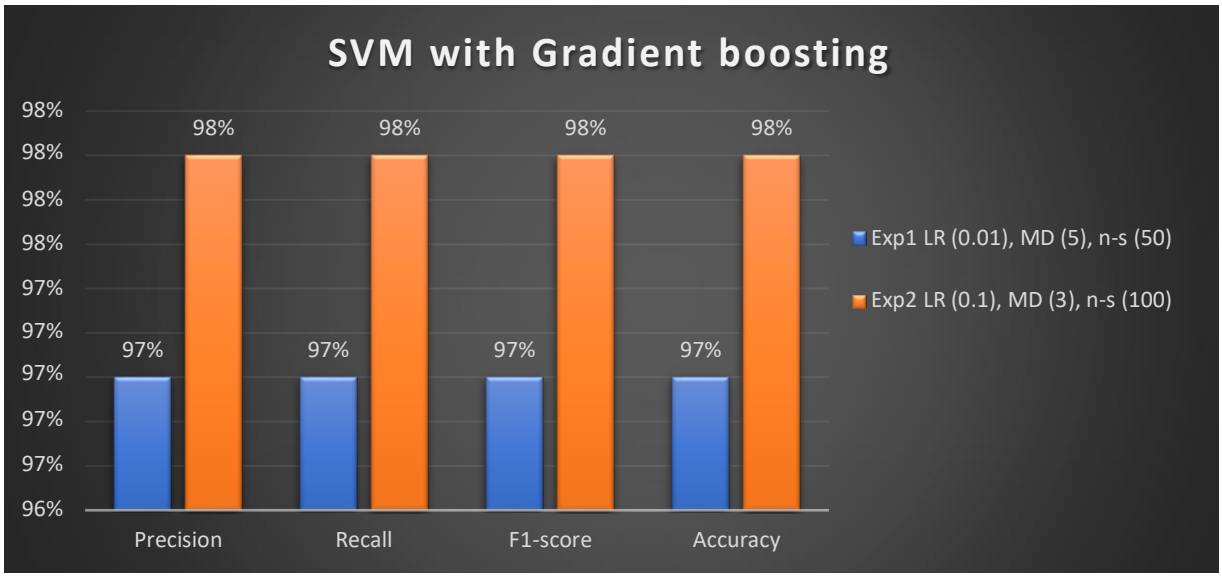


FIGURE 10 PERFORMANCE REPRESENTATION OF GRADIENT BOOSTING WITH SVM

Ensemble SVM with Bi-LSTM

Ensemble methods combine multiple models to improve overall performance by leveraging their strengths and mitigating their weaknesses. An Ensemble SVM with Bi-LSTM refers to a machine learning approach that integrates Support Vector Machine (SVM) classifiers with Bidirectional Long-Term Memory (Bi-LSTM). This thesis utilized the following parameters: 128 Bi-LSTM units, a dropout rate of 0.5, the default learning rate of the Adam optimizer, and a batch size of 32.SVM Model: Kernel type: Linear, Regularization parameter (C): 1.0 Probability estimates enabled Logistic Regression Meta-Model: Solver: lbfgs Maximum number of iterations: 1000 Regularization type: L2 penalty. The lbfgs solver in the context of logistic regression stands for Limited-memory Broyden Fletcher Goldfarb Shanno. It is a popular optimization algorithm used for solving optimization problems, particularly in the context of machine learning and numerical optimization.

4.5. EXPERIMENTATION RESULT OF DEEP LEARNING

The training and testing capabilities of LSTM and Bi-LSTM models for the categorization of idiomatic expressions have been discussed in this section. The identical hyperparameter that we used to build up the environment and hyperparameters were also used to evaluate the two models. We

constructed the suggested model once the necessary experimentation settings were completed. Using the Sklearn deep learning API, the suggested model is created. To prevent overfitting, an LSTM layer with a dropout rate of 0.5 was applied after each LSTM layer, with the first layer using 64 units and the second layer using 32 units. The model has 831 thousand parameters, all of which can be trained with a batch size of 32 across ten epochs. Ultimately, the model was trained and tested using the model development dataset, and the test accuracy was 98%.

The first Bi-LSTM layer unit in the Bi-LSTM model configuration utilizes 64, while the second one uses 32, and SoftMax is used as the activation function. Bidirectional LSTM layers' power is harnessed by the model design, which improves the model's ability to capture sequential patterns in both forward and backward directions. After every Bi-LSTM layer, dropout layers are positioned carefully to reduce overfitting and improve the model's generalization skills. The Dense layer gives the model the finishing touch and allows it to predict across numerous classes with grace and accuracy. It is embellished with the essence of SoftMax activation, and the model measure in all hyperparameters the accuracy is 98%. Trial and error outcome of the LSTM model used to identify idiomatic expressions.

Idiomatic Expression Identification Training Accuracy

FIGURE 11 TRAINING AND VALIDATION ACCURACY LSTM MODEL

Idiomatic Expression Identification Training Loss

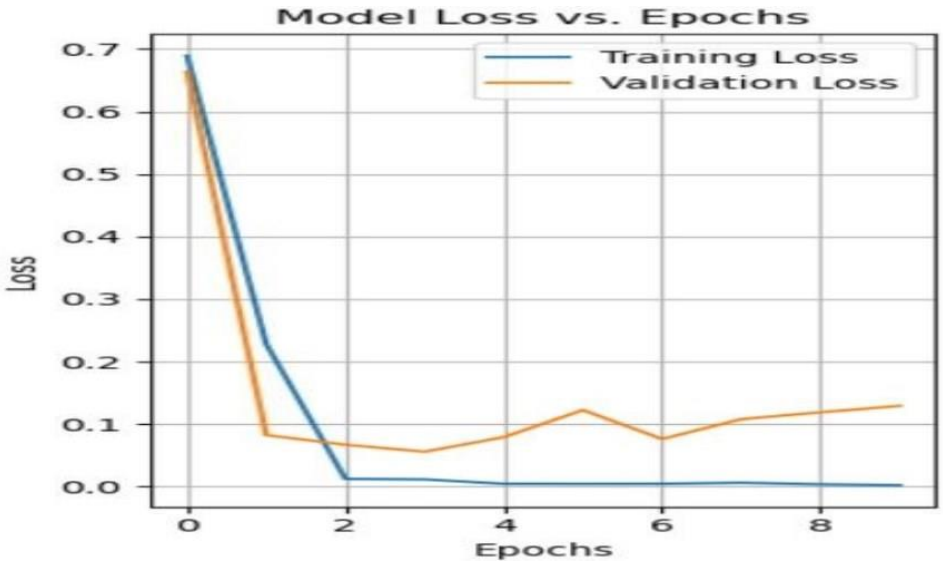


FIGURE 12 TRAINING AND VALIDATION LOSS LSTM MODEL

Experiments	precision	Recall	F1-score	Accuracy
Bs (32), Lr (0.1), Do(0.5)Ep(10)	98%	98%	98%	98%
Bs(64), Lr(0.01),Do(0.5)Ep(20)	97%	98%	97.5%	97%

TABLE 8 LSTM MODEL PERFORMANCE

Table 10 shows that in experiment one, LSTM performs well in terms of accuracy (98%), precision (98%), recall (98%), and f1-score (98%). In contrast to experiment one, the LSTM model in experiment two performs somewhat worse. The result above indicates that there is a discernible difference in the output result depending on the hyperparameter adjustment of Batch size (Bs), Learning rate (Lr), Dropout (Do), and Epoch (Ep).

Bi-LSTM Model Performance of Accuracy

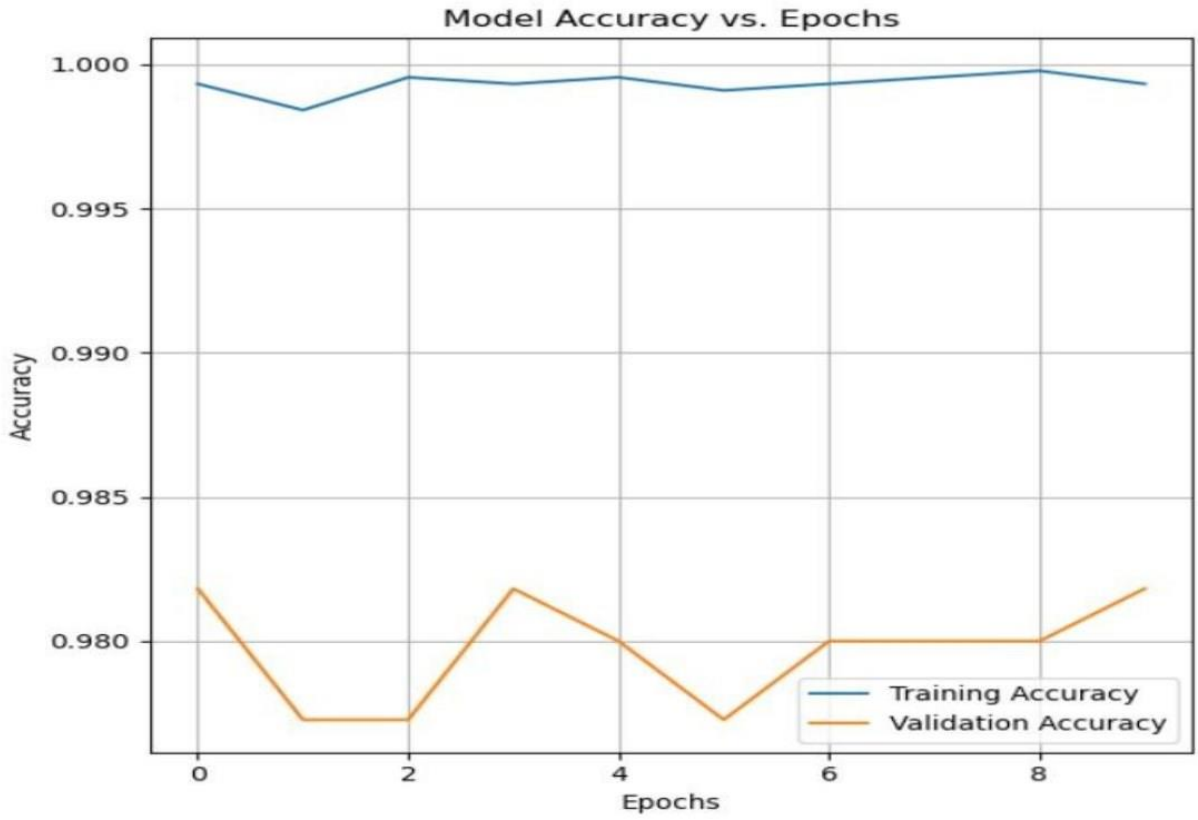


FIGURE 13 BI-LSTM PERFORMANCE OF ACCURACY

Bi-lstm training loss

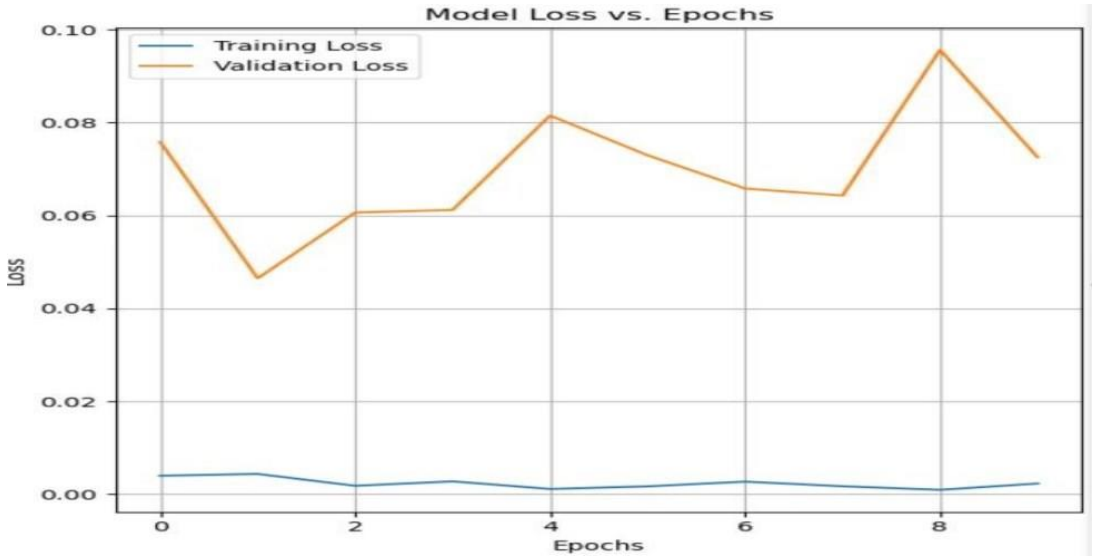


FIGURE 14 BI-LSTM TRAINING LOSS

Experiments	Precision	Recall	Fl-score	Accuracy
Bs(32), Lr(0.1),Do(0.5)Ep(10)	97.76%	97.73%	98 %	98.17%
Bs(64),Lr(0.01),Do(0.5)Ep(20)	97%	97%	97 %	97%

TABLE 9 BI-LSTM MODEL PERFORMANCE

Batch size or, Bs is the number of training samples used in each training cycle of the model. Although larger batch sizes may demand more memory, they can speed up training. The crucial factor in optimization algorithms that regulates the step size during model training is called learning rate, or Lr. Reducing the learning rate to a lower value (e.g., 0.01) results in smaller steps, which helps prevent overshooting the ideal answer dropout (Do): A regularization strategy that eliminates neurons at random during training to avoid overfitting.

The likelihood of a neuron being dropped is represented by the value. Ep (Epochs): The total number of times the model is trained using the training dataset. Although they can lengthen training time, more epochs can improve learning. Comparing Experiment 1 and Experiment 2, both show extremely high performance and comparable outcomes in terms of all parameters. This implies that for this purpose, both configurations are fairly effective. Hyperparameter Differences: The main differences between the two experiments are: Batch Size: Experiment 1 uses a smaller batch size (32) compared

to Experiment 2 (64). Learning Rate: Experiment 1 uses a higher learning rate (0.1) compared to Experiment 2 (0.01).

Epochs: Experiment 1 uses fewer epochs (10) compared to Experiment 2 (20). Therefore experiment 1 is a better result than experiment 2 because of the selected hyperparameter.

4.6. COMPARISON OF THE SELECTED MODELS

To improve the model's prediction performance, we assessed the chosen models for idiomatic phrase identification using various hyperparameter adjustments. The primary techniques for assessing the accomplished outcome are training duration and forecast accuracy. The model built during the training phase took 39.82 and 42.24 seconds for Bi-LSTM and LSTM, respectively, to complete the training. When it comes to training, LSTM requires a lot more time than Bi-LSTM. In addition to this prediction performance of 2:80 seconds and 2:79 seconds for LSTM and Bi-LSTM, respectively, the Bi-LSTM model requires less time than the LSTM model. Idiomatic Expression Identification from ML and DL Once the trained sentences have been classified as idioms and non-idioms using techniques such as SVM, Gradient boosting, LSTM, and Bi-LSTM, the task is to detect the idiomatic expression phrases from the provided text.

Comparison of models SVM, LSTM, Bi-LSTM, and ensemble models.

Model	Accuracy
SVM	98%
Gradient Boosting	65.55%
LSTM	98%
ENSEMBLE (SVM, GB)	98%
Bi-LSTM	98.17%
ENSEMBLE (SVM, Bi-LSTM)	98.27%

TABLE 10 COMPARISON USED MODEL

4.7. DISCUSSION

SVM works by finding the optimal hyperplane that separates data points of different classes in a high-dimensional space. The goal is to maximize the margin between the classes. Support Vector Machine:

A powerful and versatile model often used for classification tasks. Gradient Boosting: An ensemble method that combines with SVM to improve prediction. Long Short-Term Memory: A type of recurrent neural network particularly well-suited for sequential data, like time series or natural language processing. An ensemble method that combines predictions from multiple different models. Bi-LSTM: A bidirectional LSTM, that processes sequential data in both forward and backward directions, potentially capturing more context. Interpreting the Results: Top Performers: Bi-LSTM model ensemble with SVM and the Bi-LSTM model achieved the highest accuracy (98.27% and 98.18%, respectively). This suggests that recurrent neural networks, particularly Bi-LSTMs, are well-suited for the task at hand.SVM Performance: The SVM model also performs well (98%), indicating that it's a strong competitor for this task.Gradient Boosting: The Gradient Boosting model has significantly lower accuracy (65.55%) compared to the other models. This suggests that the task may be better suited for models that can capture complex patterns and relationships in the data, as Gradient Boosting may not be as effective in doing so. The ensemble model (SVM with GB) also achieves the highest accuracy (98.27%), demonstrating the benefit of combining multiple models.

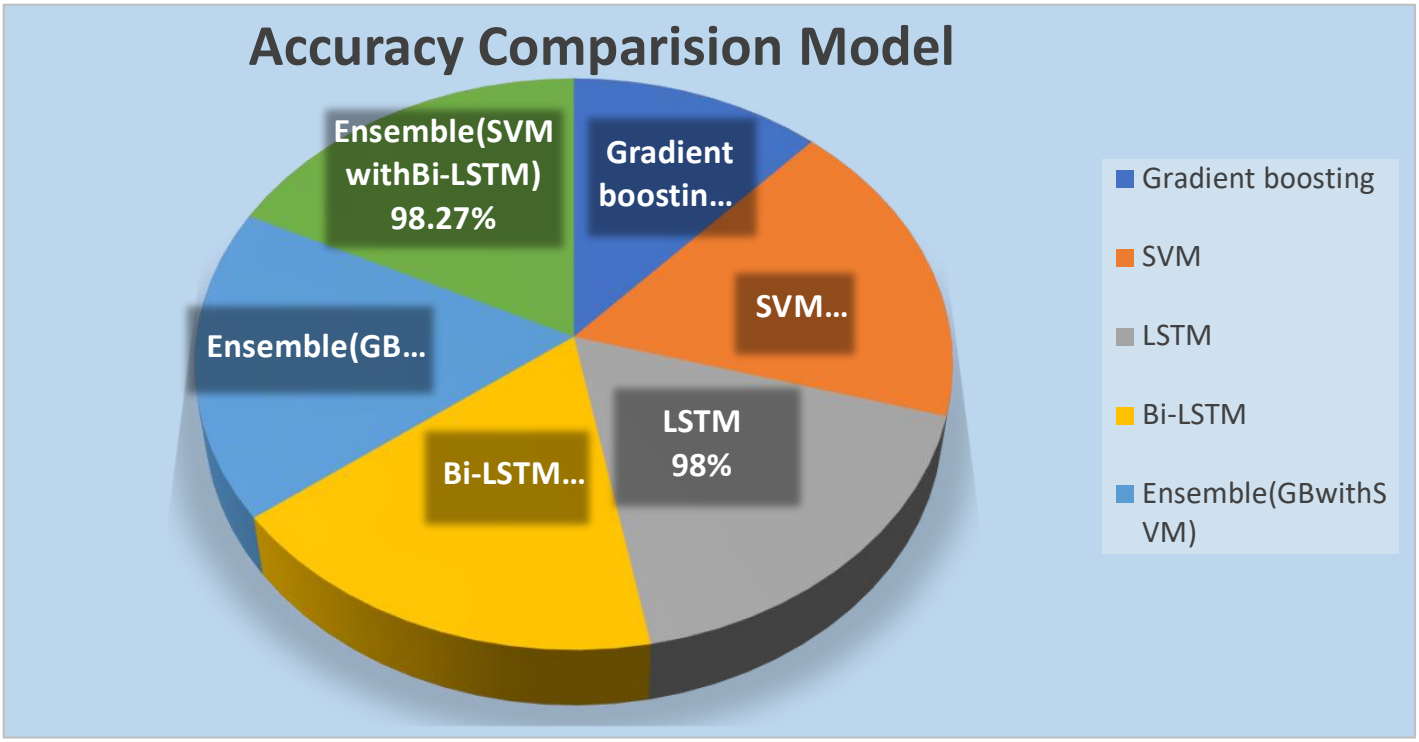


FIGURE 15 ACCURACY COMPARISON MODEL

To clean the text data in this work, URLs, non-Amharic characters, and excess whitespace are removed before the dataset is loaded from a CSV file. Rows containing NaN labels are removed once

the labels are encoded using Label Encoder. To make the text data ready for model training, it is padded and tokenized. TensorFlow's Keras API is used to build the LSTM model. To avoid overfitting, it is composed of two LSTM layers with dropout regularization after an embedding layer. The model is compiled using the Adam optimizer and categorical cross-entropy loss. It is trained using 32 batches and 10 epochs of training data. During training, the model's performance is tracked using the validation data. Using the test data, the optimal model (loaded from a stored file) is assessed. The test accuracy is recorded, and the F1 score is computed using the macro average. The classification report is produced, which displays each class's support, F1 score, precision, and recall. All things considered; the code demonstrates an organized process for classifying idioms using LSTM in TensorFlow. It manages preparing text data, training models, evaluating them, and interpreting the results with effectiveness. Two machine learning and deep learning methods have been tested for the idiomatic expression identification model. To determine which model performs better and how the various Hyperparameter setups affect the results, we examined the two models on various Hyperparameter configurations. To determine how many testing samples were correctly identified and how many were incorrectly classified for model error analysis, we also used a confusion matrix. We also looked at the effects of normalization, punctuation mark removal, and stop-word removal on the outcomes. We have employed dense layers for DL, TFIDF for ML, and Keras embedding for DL in the word representation technique. We employed support vector machines, gradient boosting techniques from machine learning, long short-term memory, and bidirectional long-term memory algorithms from the deep learning approach for experimentation.

LSTM and Bi-LSTM are two deep learning algorithms; Bi-LSTM scored higher than LSTM. The reason is that because the Bi-LSTM model employs two hidden layers, it is more context-aware than the LSTM model. The Bi-LSTM can comprehend the context of words in the input sentence because the first hidden layer learns of the sequence forward and the second hidden layer learns of the sequence backward (Kamath C. N., (2018)). SVM outperformed gradient boosting in machine learning tasks due to its kernel and threshold functions, enabling it to effectively distinguish between different data sets. Generally speaking, Bi-LSTM performs better than LSTM because it can comprehend the supplied data's forward and backward context. Our experiments have shown us that altering the hyperparameters of two algorithms alters the model's prediction performance. In conclusion, we have completed the analysis and contrasted various models' prediction performances. LSTM (98%) and Bi-LSTM (98.17%) are between the two algorithms. It is found that Bi-LSTM outperforms LSTM in idiomatic identification. Bi-LSTM outperforms LSTM because it is a powerful tool for modeling word-phrase sequential interdependence in both directions of the sequence. Overall, the bidirectional architecture allows Bi-LSTMs to leverage more information

than standard LSTMs, improving accuracy in many applications. Bidirectional Long Short-Term Memory (Bi-LSTM) networks can be more effective than regular LSTM networks for tasks like Amharic idiom identification due to several key advantages: Context Awareness, Improved Feature Extraction, Handling Ambiguity, Performance on Sequential Data, Better Generalization, Rich Representation due to this the bidirectional architecture of Bi-LSTMs provides a more comprehensive understanding of language, making them particularly suited for tasks like Amharic idiom identification, where context plays a critical role.

CHAPTER 5 CONCLUSION AND FUTURE WORK

5.1. CONCLUSION

In this study, we explored the identification of idiomatic expressions in Amharic text using four algorithms: Support Vector Machine (SVM), Gradient Boosting, Long Short-Term Memory (LSTM), and Bidirectional LSTM (Bi-LSTM). We applied various preprocessing techniques, including stop word removal, normalization, and punctuation mark removal. Preprocessing allows for better visualization and understanding of the data, which can aid in interpreting model results and making informed decisions.

For model implementation, we utilized the Keras embedding layer and dense layers for deep learning approaches, while employing TF-IDF for machine learning models. Our results demonstrated that the Bi-LSTM with SVM model significantly outperforms the other algorithms, achieving an impressive 98.27% prediction accuracy. When comparing the prediction performances, Bi-LSTM achieved accuracies of 98.27% compared to 98% for LSTM, 98% for SVM, and 98% for Gradient Boosting (SVM ensemble). Notably, BiLSTM not only provided superior accuracy but also required less training time than LSTM, thanks to its ability to effectively handle both forward and backward contextual information. Overall, the experimental results indicate that Bi-LSTM is the most effective method for idiomatic expression identification in Amharic text, outperforming SVM, Gradient Boosting, and LSTM. This research highlights the potential of deep learning techniques, particularly Bi-LSTM, in enhancing natural language processing for the Amharic language.

5.2. CONTRIBUTION OF THE STUDY

This work uses a deep learning strategy and ensemble model to construct an Amharic idiomatic identification. We have compiled a dataset of Amharic idiomatic expressions. We have 5,500 phrases ready, each phrase annotated with linguistic experts. We have created a deep-learning model that can recognize and categorize Amharic idiomatic phrases. We may conclude that the model (ensemble model) is our contribution because the morphology, grammar, and semantics of the Amharic language do not align with the one we prepared. In addition to this, the annotated data, which is ready for this thesis is used for the teaching-learning process and also the contribution of this work. identifying Amharic idioms will significantly enhance various NLP activities, such as machine translation, semantic analysis, and sentiment analysis. By incorporating idiomatic expressions, we can improve the accuracy and contextual understanding of these tasks. This, in turn, will lead to more natural and

nuanced interactions in applications like chatbots, language learning tools, and cultural content generation.

5.3. FUTURE WORK

We have created an idiomatic identification model for the Amharic language in this work. The pure or semi-pure nature of idiom words or phrases is not discussed in this work. As a result, there is still room for more research in this field. However, the current dataset may be small and, therefore produce a much larger corpus of Amharic text that is covered by a variety of domains and tagged with idioms. To make the dataset more representative of everyday language, the researcher will incorporate real-world data, such as idioms from social media, news articles, and other sources. Working together with cognitive scientists, linguists, and specialists in Amharic language and culture may yield fresh ideas and methods for addressing the problem of idiomatic identification, which may then influence the creation of more linguistically grounded and successful models. Lastly, the identification and performance of the model are improved by using BERT, Roberta, or other language understanding models.

References

[1]. A.Aklilu and D.Worku.1992. A. (1992). *Amharic Idioms*.

[2]. Abebe Fenta, A., & Gebeyehu, S. (2023). Automatic Idiom Identification Model for Amharic Language. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *22*(8). https://doi.org/10.1145/3606864

[3]. Assabie,Y (July 2021) Morphologically Annotated Amharic Text corpora

[4]. Baye Yimam. (1999). *Amharic sewasew book*.

[5]. Bzdok, D. (2018). *Support vector machine learning*.

[6]. Dagnachew Amsalu, W. (1993). የአማርኛፊሊጦች *Idiomatic expression sin Amharic*.

[7]. Endalie, D., Haile, G., & Taye, W. (2023). Deep learning-based idiomatic expression recognition for the Amharic language. *PLoS ONE*, *18*(12 December). https://doi.org/10.1371/journal.pone.0295339

[8]. Fissaha, S., & Haller, J. (n.d.). *Application of corpus-based techniques to Amharic texts*. http://www.iai.uni-sb.*de*

[9]. *Gasser Michael. (n.d.). horn Morpho. A system for morphological processing Amharic, Oromo, And Tigrinya*,

[10]. Haddis Alemayehu. (1996). *FIKIR ESKEMEQABIR*.

[11]. Hinkel, E. (2017). Teaching Speaking in Integrated-Skills Classes. In *The TESOL Encyclopedia of English Language Teaching* (pp. 1–6). Wiley. https://doi.org/10.1002/9781118784235.eelt0256

[12]. Kamath C. N. (2018). *Deep learning*.

[13]. Kelemework, W. (2013). Automatic Amharic text news classification: A neural networks approach. In *J. Sci. & Technol* (Vol. 6, Issue 2).

[14]. Kuncoro, A., Dyer, C., Hale, J., Yogatama, D., Clark, S., & Blunsom, P. (n.d.). *LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better*. Association for computational linguistics. https://github.com/tensorflow/models/

[15]. Mäntylä, K. (n.d.). *Idioms and language users: the effect of the characteristics of idioms on their recognition and interpretation by native and non-native speakers of English*. https://www.researchgate.net/publication/277872357

[16]. Melamud, O., Goldberger, J., & Dagan, I. (n.d.). *context2vec: Learning Generic Context Embedding with Bidirectional LSTM*. http://www.cs.biu.ac.il/nlp/resources/

[17]. Muzny, G., & Zettlemoyer, L. (n.d.). *Automatic Idiom Identification in Wiktionary*. Association for Computational Linguistics.

[18]. Peng, J., Feldman, A., & Vylomova, E. (n.d.). *Classifying Idiomatic and Literal Expressions Using Topic Models and Intensity of Emotions*.

[19]. Rokach, L. and M. O. (n.d.). Function and use of Decision Tree. *2005*.

[20]. Seid Muhie Yimam, H. M. A. A. A. and C. Biemann. (2021). *Exploring Amharic sentiment analysis from social media texts: Building annotation tools and classification models*.

[21]. Shahzad, M. F., Xu, S., Lim, W. M., Yang, X., & Khan, Q. R. (2024). Artificial intelligence and social media on academic performance and mental well-being: Student perceptions of positive impact in the age of smart learning. *Heliyon*, *10*(8). https://doi.org/10.1016/j.heliyon.2024.e29523

[22]. Sileshi Girmaw Miretie. (n.d.). Amharic stop words. *2018*.

[23]. Tewodros Hailemeskel. (2003). *Amharic punctuation mark*.

Appendixes

Appendix A Sample dataset

ውሃወቀጣ,ውጤትየሌለው አይነውሃ,ሁኔታ ውሃአጣጭ,የትዳርአጋር እጅአጠረው,ቸገረው ወፍዘራሽ ,ዘሩየማይታወቅ የደምገንቦ,ቆንጆ እብርትየለሽ,ሆዳም ማርምአልስ,ጠግቤአለሁ ልቡሰባ,ተበተ አልሞትባይተጋዳይ ,ራስንመከላከል አደብገዛ,ጥሩባህሪይአሳየ ሹልአፍ,ነገረኛሽክሜወረደልኝ,ሀሳቤቀለለከልኩአያልፍም,የሆነውይሆናል ጆሮ ጠቢ ወሬኛ ቅስሙተሰበረ,ተአስፋቆረጠ ቢከፍቱትተልባ,ምንምየሌለው ሰኔናሰኞ ,ግጥምጥሞሽ ጎጆወጣች,ቤትሰራች ልበ ሰፊ ክንዱንተንተራሰ,ሞተ ጆሮጠቢ ,ሰላይ መለስቀለስ,ተመላለሰ በደረቁላጨ,አታለለ ግምባራም,እድለኛ አይነሌባ,ሴሰኛ ሲበሉየላኩት,ልበቢስ የቀለምቀንድ,ምሁር ልቅምያለች,ቆንጆ የውሻቁስል,ቀላል የእርጎዝንብ,ጥልቅየሚል ራሱን ቻለ አይነልም,ተንኮለኛ ፍርደገምድል,የተዛባፍርድ የእንቧይካብ,በቀላሉየሚፈርስ አንጀትበላች,አሳዘነች ዝባዝንኬ የማይረባ በሬወለደ,የውሸትወሬ መሬትቁናሆነ,ተሸበረ ሆድናጀርባ,አለመግባባት ሰውሆነ,ራሱንቻለ ስምህንቁስይጥራው,ሙት ውርድከራሴ,ሀላፊነትየለብኝም ሀረግመምዘዝ,ዘርመቁጠር ብቅልአውራጅ,እረጅም ምራቁንየዋጠ,በሰልያለ መሬት ላሰ የህልምሩጫ,ምኞት ሆደባሻ,ቶሎየሚከፋው የቤትልጅ,ጨዋ ቅቤጠባሽ,አዛኝመሳይ ግንባርአስመታ,ፊቱንአሳየ ቆርጦቀጥል,ውሸታም ጨርቋንጣለች,አበደች ቆሞቀር,ያላገባ ሮጦያልጠገበ,ወጣት ጅብፍቅር,ቀንሲበላሽየሚጠፋ ወምበርገፍ,ተቀያሪ ሆዱቆረጠ,ጨከነ ሰማዩተደፋበት,ግራገባው ሆደሰፊ,ታጋሽ እዩኙእዩኝባይ ወሬኛ ሰማይ ጠቀስ ጥልአበቀለ,ጥልአነሳጥልያለሽዳቦ,ጥልየሚሻጥሩደሀ,ሀብትየሌለውጥሬሰው,ያልተማጥሬስጋውንበሉተ,አሙትጥሬነገር,አ ዲስነገርመንገድይጠቀለልለታል,ያፈጥነዋልሆዱንአለበው,አስቀመጠው አለብላቢትምላስ,ክፉተናጋሪ ከተማ ቀመስ ስልጡንአለንጋጣት,ባለረጅምጣትአላምጠውየተፉት,ንቀውየተውትባልአልጋ,ንጉመጻሕፍትአመሰካከረ,አወዳደረሆደማ ጭድ,ክፉሰውፈረንጅ,ስልጡን ጎጆውፈረሰ,ትዳሩፈረሰ ፍርደገምደል,አድሎዋዊ ጸሀይወጣ,እውነቱተገለጠ የሸኮኮጸሎት,መጨረሻውየማያምርየስጋቁራጭ,የቅርብዘመድጭዳአለ,ሰውገደለጭቃአፍ,ዝምብሎየሚናገርጭራውንአ ማታ,ተለማመጠነገርጫረ,ጠብጀመረየጨረባተዝካር,ውካታጥጃሆድ,ትንሽሆድከርሞጥጃ,የማይሻሻልየሞትጥዋ,የሞትተ ራስሜቱቀዘቀዘ,ቅስሙተሰበረወኔውቀዘቀዘ,ወኔውበረደቅዝዝአለው,ቅፍፍአለውልብሷንቀደደች,በጣምአዘነችቅድአፍ,ወ ሬኛቆቡንቀደደ,ምንኩስናውንአፈረሰጉህቀደደ,ፀሐይወጣቀጠፈ,ዋሸባጭርተቀጠፈ,በልጅነቱሞተኮሶተጣባው,ኮሶታየው ጆሮጠቢ,ሰላይጠቦትፊት,መልኩየሚያምርደመናውተጠንስሷል,ዝናብሊዘንብጥንቡንጣለረከሰወሬውጠነዛ,አልማርክአለ ማህበርተጣጣ,ተዋደደጢንአለ,ተኩራራልብሱንጣለ,አበደልሷንጥላበታለች,በጣምወዳዋለችጠጅጣለ,ጠጅጠመቀጣጣ ቴ,ተናግሮየማያቆም መጣፍገላጭ,ጠንቋይ ጣፋርዬ,ጋጠወጥ ጐመንበጤና,ከነገርራቀ ,ልጓምጣሰ ህግ አፈረሰ ሎሚ በየተራ ሁሉን አዳራሽ ምድር ለቀቀ ነጋ ምድር ያዘ ምጣዱሰማ ተዘጋጀ ሳቅ አነቀው ሊስቅ ነው ሰማይ ተደፋበት

ተጨነቀ ሲበሉየላኩት ችኩል ጎጆወጣች ተዳረች ጭቃ አፍ መጥፎ ተናጋሪ ጎተራ ሆድ የማይጠግብ ግንባረደረቅ እድለ ቢስ ውሃ ስንቁ ምግብየማይበላ ወጥ ረገጠ ተዳፈረ ወሬምሳው ወሬኛ ኩቴደርቅ እድለቢስ በደረቁ ላጨ ዋሸ እሾህን በእሾህ ተመሳሳይ ነገር እሳት ሆኗል ተወዷል አይነ ልም ተንኮለኛ የእንቧይ ካብ መሰረተ ቢስ አንጀት በላች አሳዘነች

Appendix B Model code

```
# LSTM Model Configuration
model = Sequential()
model.add(Embedding(input_dim=len(tokenizer.word_index) + 1, output_dim=128, input_length=max_length))
model.add(LSTM(64, return_sequences=True))
model.add(Dropout(0.5))
model.add(LSTM(32))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))  # Adjust for the number of classes

# Model Building for multi-class classification with two BiLSTM layers
model = Sequential()
model.add(Embedding(input_dim=len(tokenizer.word_index) + 1, output_dim=100, input_length=max_length))
model.add(Bidirectional(LSTM(128, return_sequences=True)))
model.add(Dropout(0.5))  # Dropout layer
model.add(Bidirectional(LSTM(128)))  # Increased units
model.add(Dense(num_classes, activation='softmax'))  # Output layer for multi-class classification
```

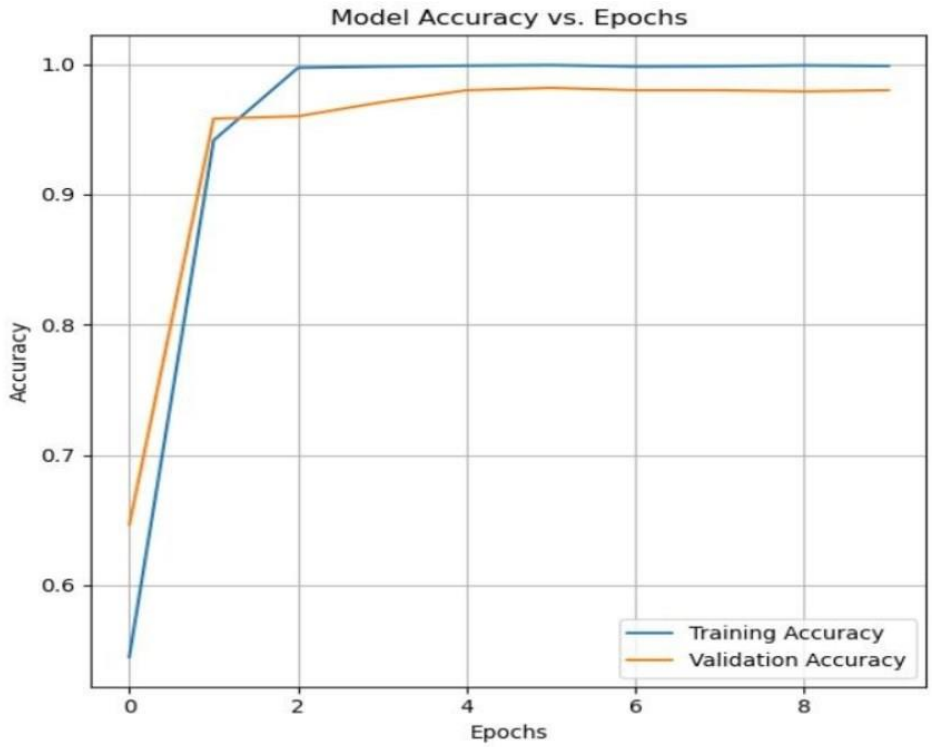
```
# Create a Voting Classifier
voting_model = VotingClassifier(estimators=[
    ('svm', svm_model),
    ('dt', decision_tree_model)
], voting='soft')  # Use soft voting to consider predicted probabilities

# Define the parameter grid for Grid Search
param_grid = {
    'svm__C': [ 10],  # SVM regularization parameter
    'svm__gamma': [1],  # SVM kernel coefficient
    'dt__max_depth': [None]  # Decision Tree maximum depth
}

# Gradient Boosting Model Configuration with Grid Search
grid_search = GridSearchCV(voting_model, param_grid, refit=True, verbose=3)

# Train the Voting Classifier with Grid Search
```

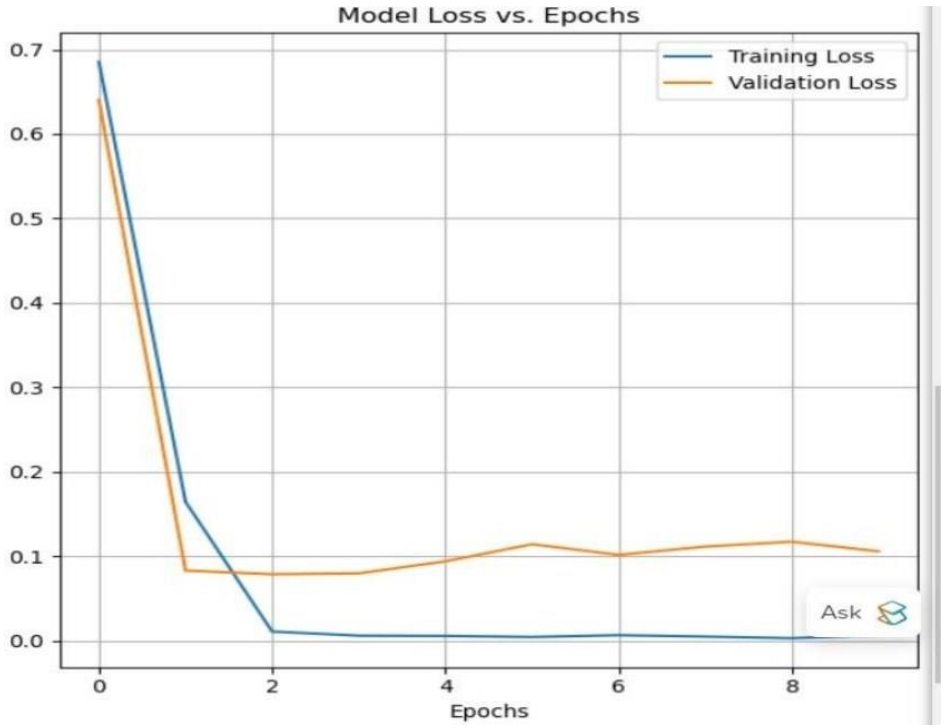
Appendix C Idiom Identification

```
# Function to predict the label of a user-input expression
def predict_expression(user_expression):
    cleaned_expression = clean_text(user_expression)
    sequence = tokenizer.texts_to_sequences([cleaned_expression])
    padded_sequence = pad_sequences(sequence, maxlen=max_length)
    prediction = loaded_model.predict(padded_sequence)
    predicted_class = np.argmax(prediction, axis=1)[0]
    predicted_label = label_encoder.inverse_transform([predicted_class])[0]
    return predicted_label

while True:
    # Get user input
    user_input = input("Enter an expression (or type 'quit' to exit): ")

    if user_input.lower() == 'quit':
        break

    # Predict the label
    predicted_label = predict_expression(user_input)

    # Print the predicted label
    if predicted_label:
        print(f"Predicted label: {predicted_label}")
    else:
        print("No matching word found in the database.")
```

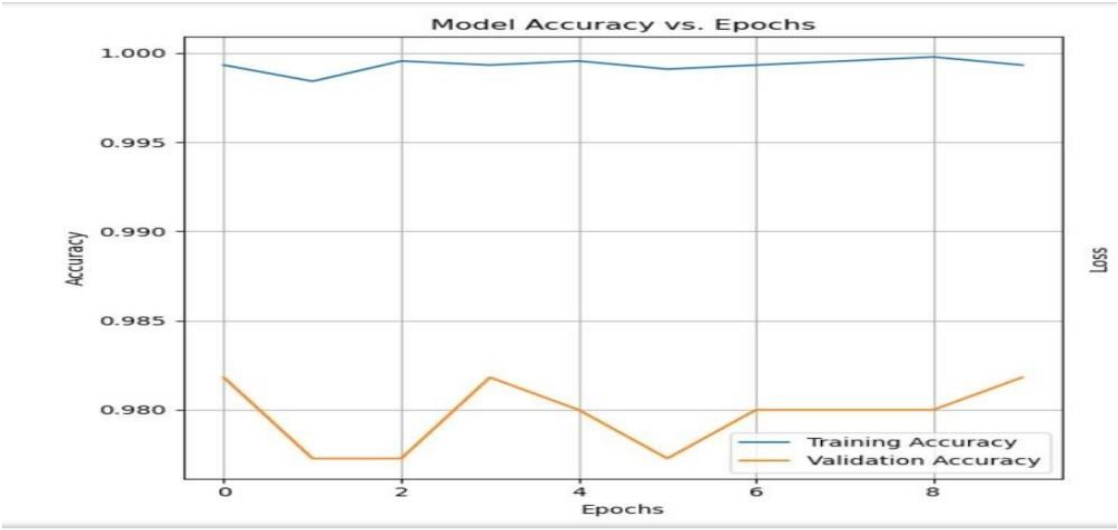
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step
Predicted label: idiom
Enter an expression (or type 'quit' to exit):  አልሞትባይተጋዳይ
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
Predicted label: idiom
Enter an expression (or type 'quit' to exit):  ሾልለፍ
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
Predicted label: idiom
Enter an expression (or type 'quit' to exit):  ብረትድስት
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 68ms/step
Predicted label: literal
Enter an expression (or type 'quit' to exit):  ወንበር
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step
Predicted label: literal
Enter an expression (or type 'quit' to exit):  ↑↓ for history. Search history with c-↑/c-↓
```

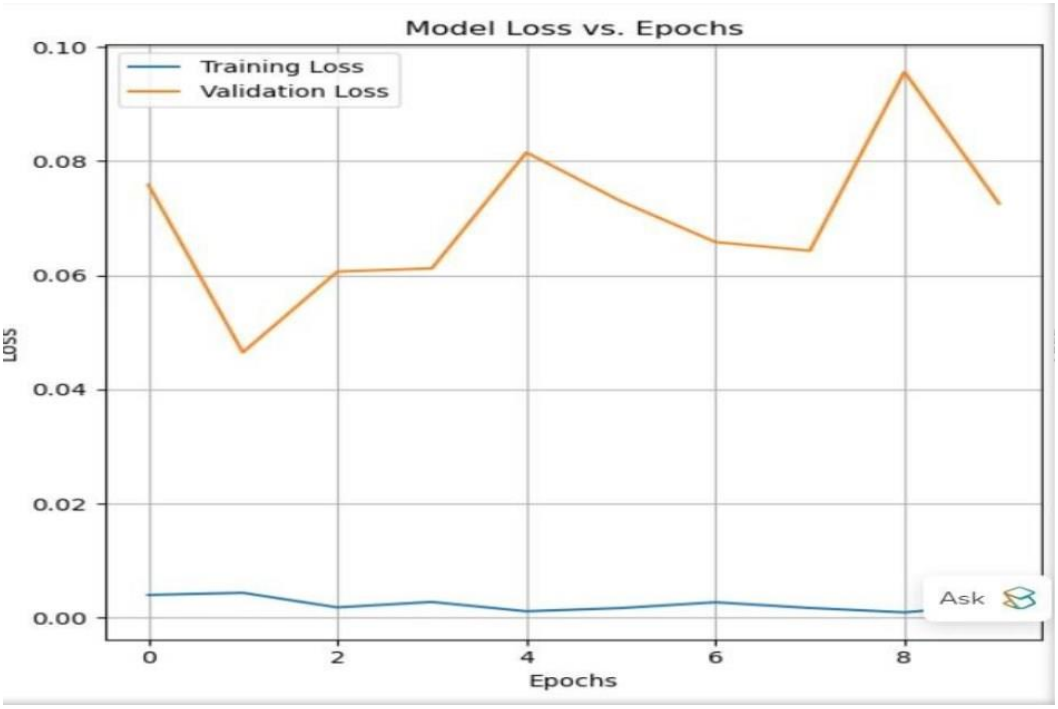
Appendix D LSTM MODEL



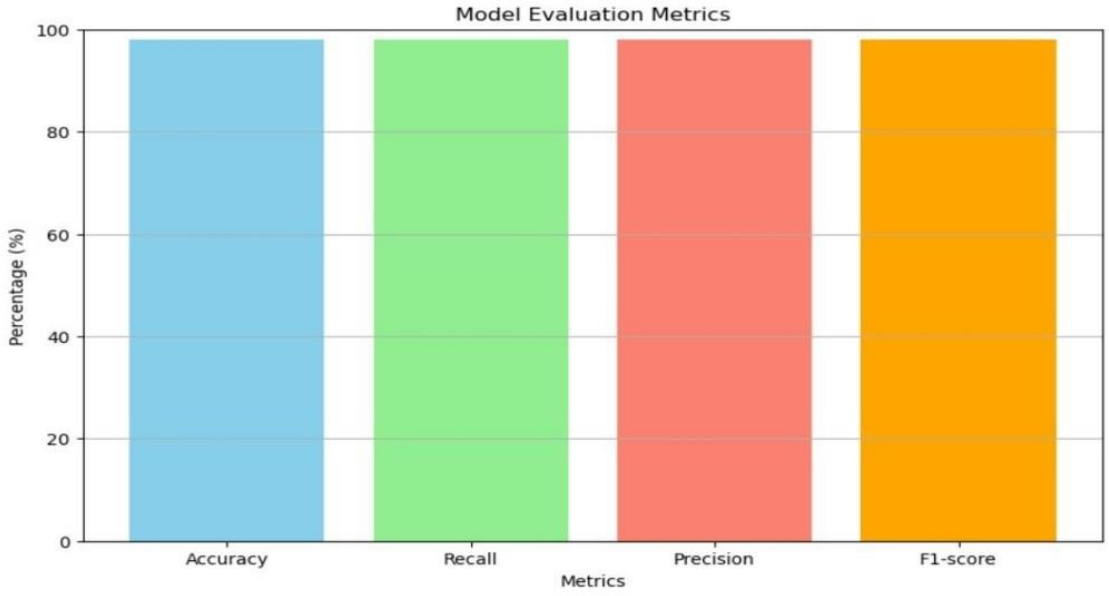


Appendix E Bi-LSTM Model

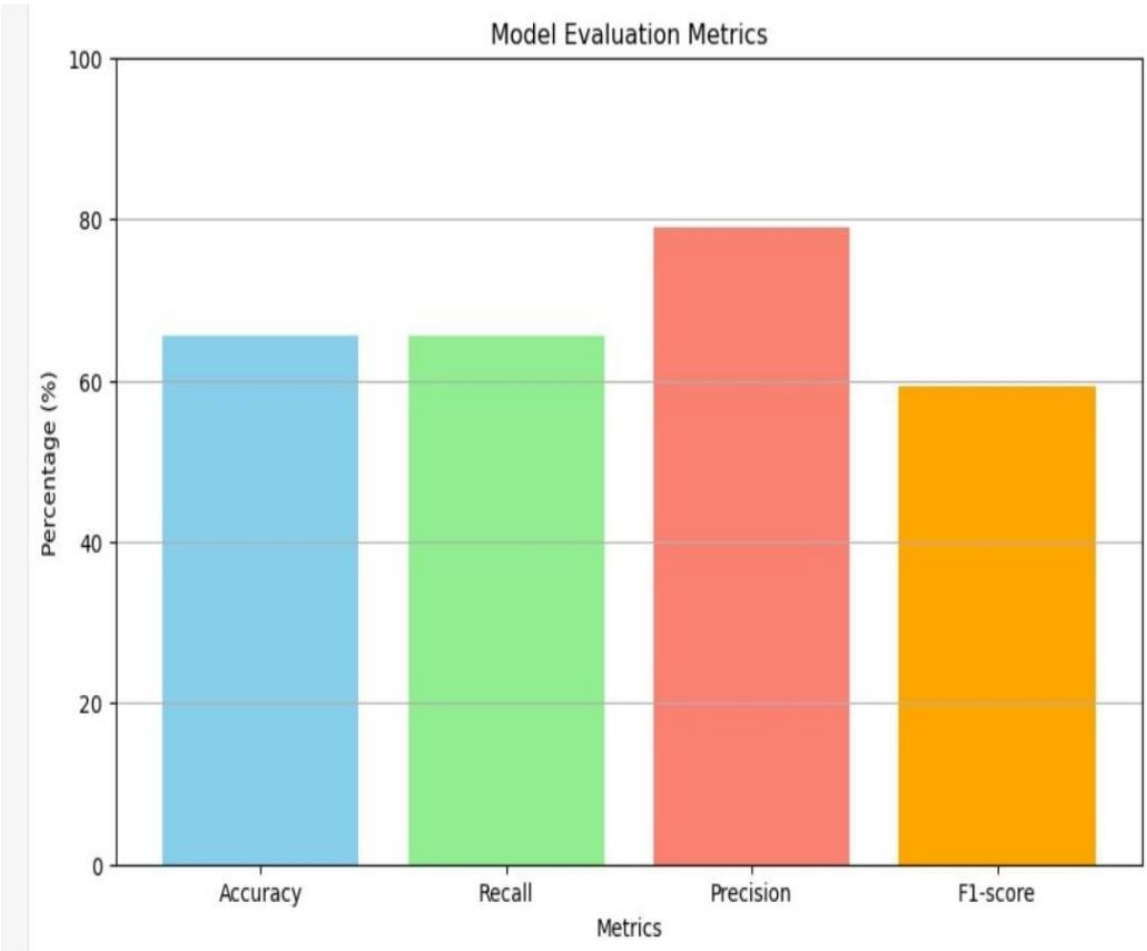




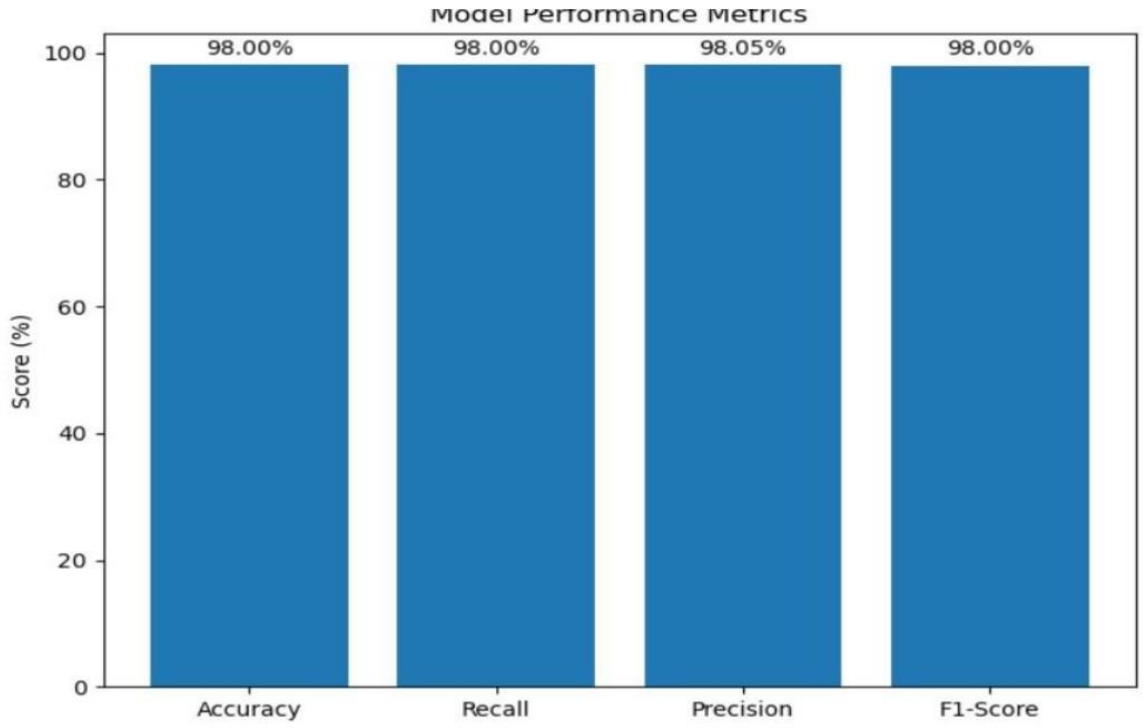
Appendix F SVM model



Gradient Model



Appendix G Ensemble Model



Appendix H comparisons Model