DMU

Advanced Journal of Interdisciplinary Studies 9(2025) 2303-2319

AJIDS

Journal homepage: www.ajids.com.et

Volume 9(1), June 2025

Sentiment Classification from Social Media Amharic Text Using Deep Neural Network Approach: Politics Domain

Ewunetu Birhanie*, Zewdie Mossie, Minalu Chalie

Department of Information Technology, Debre Markos University, Debre Markos, Ethiopia

*Corresponding authors Email: ewunetu_birhanie@dmu.edu.et / ewunetut121@gmail.com

Abstract

Political parties utilize social media platforms to transparently disseminate their policies, development plans, and strategies to the public. This approach encourages social media users and voters to actively engage by responding to and providing feedback on these political issues. As a result, political parties can consider the public feedback as an input for making different decisions according to the public comments and ideas with respect to their governmental policy and strategies. However, manually assessing those massive messages and feedbacks in order to make informed decisions is exceedingly tough. Because reading or reviewing such a big number of comments and reviews using standard methods is difficult and time consuming. As a result, sentiment classification, or the computational analysis of people's thoughts, attitudes, and opinions about a certain subject, has received a lot of interest. In this thesis, we used deep learning techniques such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bidirectional Long Short-Term Memory (BiLSTM) to develop sentiment classification model. In addition, word embedding methods like Word2Vec and FastText were examined to improve classification performance. The goal of this thesis is to use deep learning approaches to classify Amharic social media political posts and comments into Positive, Negative, and Neutral, making the sentiment classification problem effective. We compared various deep learning algorithms and word embedding approaches in order to determine which is the best. The proposed method able to achieve increased accuracy, precision, recall, and F1-score with 0.84, 0.92, 0.86, and 0.89, respectively, utilizing BiLSTM with FastText embedding. This promising result can help political parties and voters in quickly analyzing various opinions by reducing the amount of time and effort required to read these posts and comments.

Keywords: Deep learning approach, FastText, Sentiment classification, Word2vec

1. Introduction

Social media platforms enable individuals to openly communicate their thoughts and ideas, facilitating swift information sharing among users. Users utilize these platforms to share messages, comments, photos, videos, and various details regarding their daily lives or other relevant topics [1]. Various social media platforms, such as Facebook, Twitter, and YouTube, provide convenient and efficient ways of communicating and

sharing information publicly. For this reason, the number of social media users increased over time, and there is a high volume of textual data on the internet. As a result, the automatic extraction of relevant information from large amounts of unstructured text has attracted the interest of many researchers in a variety of fields, particularly in the NLP community [2]. In the real world, political parties are always looking for suggestions or public opinion to better understand public contentment with the policies and tactics they build for better decision-making or better serving the public. In the past, political parties used to conduct surveys or gather opinions in person to gauge public satisfaction with governmental issues. However, this method was time-consuming and expensive. Nowadays, politicians are increasingly establishing official online platforms to engage with the public[3]. Citizens also use social media to provide feedback on government performance, the implementation of social welfare schemes, and public sector services. So, before making a voting decision in political action, analyzing political party plans and development is very important, especially for voters, in order to give a vote for a better political party based on their plans and strategies that are available on different political party pages and according to other people feedbacks towards the issue. But analyzing these many posts and feedbacks manually is very difficult for political parties as well as the public for decision-making. Because, reading or analyzing so many opinions and reviews using traditional techniques is inadequate and time-consuming [4].

Hence, in order to address this issue, researchers have created a sentiment classification system for Amharic social media text using deep learning methods within the realm of politics. Sentiment classification involves the computational analysis of individuals' opinions, sentiments, emotions, appraisals, and attitudes towards various entities, including products, services, organizations, individuals, issues, events, topics, and their characteristics[5]. The goal of sentiment analysis is to analyze people’s opinions in such a way that they can help in making decisions or improving the working principles, policies, and strategies of a political party.

A few researches have been done in the area of sentiment analysis of Amharic language using machine learning as well as deep learning. Amharic sentiment analysis using deep learning approaches proposed by [6],used count vectorizer and TF-IDF for feature extraction and multi-layer perceptron (ANN) . However, these feature extraction techniques, do not keep semantic information. In addition, multi-layer perceptron cannot capture sequential information in the input data. For this reason, these feature extraction techniques and algorithms used by the researchers are not suitable, and we need other ways of feature extraction techniques and algorithms that are suitable for sequential data. Sentiment analysis of social media Amharic text using Word2Vec proposed by [7],the researchers used deep learning algorithms and word2vec for feature extraction techniques. But, this technique creates out of vocabulary problem [8], for some words that

are not present in the training corpus, and did not incorporate sarcasm expressions.

Since, many social media sites and pages are populated by sarcasm expressions. Because, Most of the time, social media users write their comments and posts by incorporating sarcasm expressions, especially in the political domain. As a result, the main aim of the author in this research is performing sentiment classification for social media Amharic texts in politics domain by involving sarcasm expression to minimize incorrect sentiment classification that was not done by previous researchers.

Sarcasm is a non-literal expression that uses positive expressions to present negative meanings [9]. This unique form of expression involves using words in a way that contradicts the intended meaning, often for the purpose of insulting, mocking, expressing anger, or humor. In the realm of communication, sarcasm is characterized by the speaker delivering their message indirectly, with the content usually carrying a concealed or underlying significance[10]. Unlike simple negatives, sarcastic sentences use positive words or reinforce positive words to convey negative opinions [11]. According to [12], there are mostly two types of situation for sarcasm occurring in the text. They are

(i) When the text contradicts a fact or fact negation.
(ii) When the text sentiment conflict with text situation: - which means using only positive sentiment or words for negative situation

This particular study effectively classifies Amharic social media political content into Positive, Negative, and Neutral sentiments by incorporating sarcasm expression, a facet overlooked by previous researchers, through the use of deep learning techniques like LSTM, GRU, and BiLSTM, in conjunction with word embedding methods such as Word2Vec and FastText. By comparing various algorithms and embeddings, the study achieved notable improvements in accuracy, precision, recall, and F1-score, especially with BiLSTM and FastText. This approach's success can streamline the analysis of diverse opinions for political parties and voters, saving time and effort typically required for manual reviews.

The contributions of the author in this research is listed below.

- We developed Amharic labeled dataset for sentiment classification from three public social media like Facebook, Twitter and YouTube.
- We prepared a pre-trained FastText and word2vec word embedding model.
- We designed and developed a sentiment classification model in politics domain that involves sarcasm expressions.
- Developing sample prototype for Amharic sentiment classification in politics domain.
- We investigate the performance of different deep learning algorithms with different word embedding techniques like FastText and word2vec for Amharic text sentiment classification in terms of accuracy, precision, recall, and f1-score and confusion matrix.

2. Literature review

The author performed a sentiment analysis on Twitter data within the political sphere. Initially, Twitter data concerning politics was gathered from the public. Word embedding methods were applied to translate the textual data into a multidimensional numerical vector structure. A dataset comprising 4400 tweets regarding Indonesian politics was employed for model training. The gathered information underwent preprocessing and conversion into numerical vectors. Subsequently, these vectors were fed into a trained recurrent neural network model for prediction purposes, resulting in an accuracy rate of 85%[13].

In their study, researchers [14] enhanced sentiment analysis for the Arabic language by leveraging word representation. The dataset was compiled from ten newspapers originating from eight distinct Arabic nations, allowing for broad coverage of words across different Arabic dialects. To boost system performance, several preprocessing tasks were executed. The researchers employed word2vec embedding methods to generate an Arabic vocabulary. They gathered a total of 2026 datasets from Twitter, which were annotated by three human annotators. Ultimately, the experimental findings demonstrated that they achieved an accuracy rate of 92% on their openly available Arabic-language healthcare dataset.

In the study proposed by [15], the authors recommend employing a combination of convolutional neural networks and bidirectional long short-term memory (CNN-BiLSTM) for sentiment analysis in Afan Oromo. Data was gathered from Facebook and Twitter social media platforms, followed by various preprocessing steps to enhance dataset organization. The refined data underwent manual annotation by four distinct annotators, categorizing it into five groups: very positive, positive, very negative, negative, and neutral. Subsequently, utilizing convolutional neural networks, bidirectional long short-term memory, and a fusion of convolutional neural network-bidirectional long short-term memory with character-level word embedding, the researchers conducted diverse experiments on the refined Facebook corpus. By splitting the dataset into 80% training and 20% testing subsets, the researchers achieved promising performance accuracies of 93.3%, 91.4%, and 94.1% for CNN, Bi-LSTM, and CNN-Bi-LSTM, respectively, based on the Facebook dataset.

In a study focusing on sentiment analysis of Amharic text opinions using deep learning methods [2], researchers gathered 1600 reviews from the official Facebook page of Fan Broadcasting Company. The data was collected via the Graph Application interface of Facebook and pertained to topics such as immigration, war, and public relations. These reviews contained text data as well as emojis and were categorized into seven classes (positive, very positive, extremely positive, neutral, negative, very negative, and extremely negative) through annotation by linguistic experts. The researchers employed deep learning algorithms, specifically artificial neural networks (ANNs). They utilized both count vectorizer and TF-IDF vectorizer for feature extraction, with experimental results

indicating superior performance for the count vectorizer over TF-IDF. Following various experiments, an average training accuracy of 70.1% was achieved.

In their study [7] , researchers conducted sentiment analysis on Amharic social media text using word2vec. Data was collected from platforms like Facebook and Twitter between February 2019 and July 2019 across various domains. A total of 11,200 data points were gathered and subjected to diverse data processing techniques. Following text cleaning procedures, the data was annotated into three categories: positive, negative, and neutral. The researchers utilized word2vec to construct the vocabulary and employed deep learning models including LSTM, GRU, and BiLSTM, conducting multiple experiments. By evaluating the performance of these deep learning algorithms with word2vec word embedding, they achieved accuracies of 84%, 86%, and 88% for LSTM, GRU, and BiLSTM, respectively.

In their investigation [16], researchers delved into Amharic sentiment analysis extracted from social media text, sourcing datasets from Facebook and Twitter within the Ethiopic Twitter Dataset for Amharic (ETD-AM). They amassed over 9.4k tweets, each annotated by three distinct users across four categories: positive, negative, neutral, and mixed. Employing word2vec, the researchers constructed various deep learning models for sentiment classification. For word2vec-based word representations, approximately 15 million sentences were collected from diverse origins such as a News dataset via the Scrapy Python API, YouTube comments through the YouTube Data API, and a Twitter dataset using the Twitter API. Leveraging a logistic regression machine learning algorithm, an accuracy of 58.42% was attained. Meanwhile, employing a support vector machine led to an accuracy of 53.19%. For fast text word embedding, an accuracy of 54.12% was achieved, and with word2vec, the researchers secured an accuracy of 55.40%.

3. Proposed system architecture

The overall architecture of the proposed sentiment classification model is depicted in **Figure** *1*.

4. Dataset

The dataset was collected from January 2021 to September 2021 from different discussion group pages and media; we considered only posts and comments on political issues. The dataset contains 3753, 3588, and 2242 negative, positive, and neutral comments, respectively, for a total of 9583 labeled comments. We used the total corpus size of 446214 collected from social media in different domains to generate vocabulary using word2vec and FastText. After that, before feeding the prepared dataset to the deep learning algorithm for training, we split the dataset into training and testing sets. To do this, we used the train_test_split module provided by the sickit-learn library.

Therefore, in this study, we used approximately 90% of the dataset for training and 10% for testing.

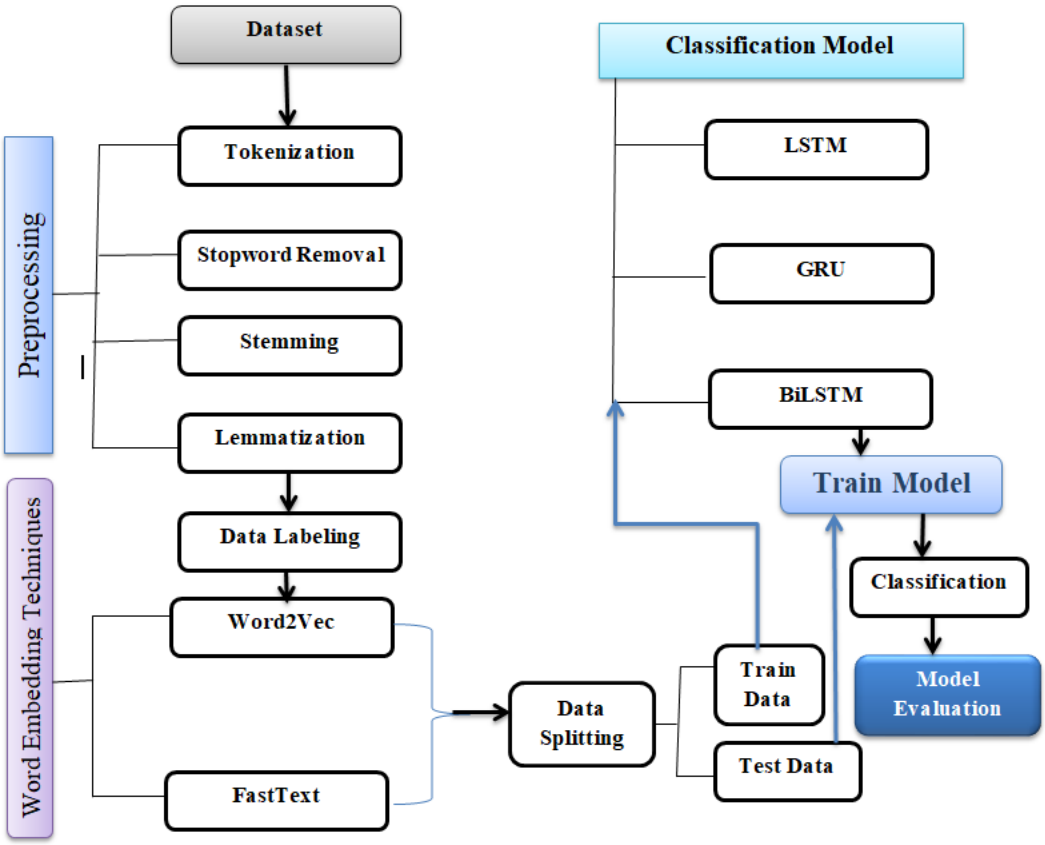

Figure 1. Architectural Design for Amharic Sentiment Classification

The
Table *1* illustrates three categories of sentiment classification involving class, label, and the number of comments:

5. Preprocessing Techniques

Data preprocessing is a basic procedure for building a machine or deep learning model. When the data are fairly preprocessed, the results are reliable. Amharic, being a morphologically rich language, requires better text preprocessing techniques to achieve better results in sentiment classification. The preprocessing component is used for cleaning the input texts for further analysis. Preprocessing activity involves different tasks, such as data cleaning, tokenization, stop-word removal, normalization and Stemming and Lemmatization.

Data cleaning should be properly performed to improve classification models. Therefore, in the text cleaning stage, we performed initial cleaning or removal of nonstandard words such as null values, empty spaces, digits, punctuation marks, expanded abbreviations to their long forms, HTML, URLs, and non-Amharic characters. Stop words are commonly used words in a language that carry little to no meaningful

information for tasks like text analysis or natural language processing (NLP).

Table 1. Three Categories of Sentiment Classification

Class	Label	Number of Comments
Positive	1	3588
Negative	2	3753
Neutral	0	2242
Total		**9583**

They are typically filtered out during preprocessing to focus on more significant words that contribute to the specific task, such as sentiment analysis or text classification. Below is a partial list of commonly used Amharic stop words that are often removed during NLP preprocessing tasks [17].

Pronouns: እኔ (I), አንተ (you, masculine), አንቺ (you, feminine), እሱ (he), እሷ (she), እኛ (we), እነሱ (they)

Conjunctions: እና (and), ወይም (or), እምነት (but), እንዲሁም (also)

Prepositions ከ (from/of), በ (in/at), ወደ (to), ላይ (on), እስከ (until)

Articles/Determiners ይህ (this, masculine), ይሄ (this, feminine), እነዚህ (these), እንደዚህ (like this)

Auxiliary Verbs ነኝ (am), ነው (is), ነን (are), ነበር (was)

Question Words ምን (what), ማን (who), እንዴት (how), መቼ (when), የት (where), ለምን (why)

Adverbs በጣም (very), እንደገና (again), እስከአሁን (until now)

Particles እንደ (like), የሆነ (which), እንበለን (let us)

Tokenization is the process of splitting a sentence into a list of words. The document is divided into sentences, and a sentence is further divided into words. In our case, the smallest piece or token is known as a word. Amharic words frequently include prefixes, infixes, and suffixes for grammatical functions, which makes tokenization more challenging. However, we use NLTK and rule-based tokenization techniques that uses predefined linguistic rules to separate words into meaningful components. And Morphological analyzer tools like HornMorpho or adaptable tokenizers such as spaCy with Amharic-specific rules can handle this complexity. HornMorpho breaks down the sentence into its individual components (words) based on the morphology of Amharic. It accounts for prefixes, suffixes, and root words, making tokenization accurate for the language's structure. Example "ከቤተሰብህን" → ["ከ", "ቤተሰብ", "ህን"]. For example, when the sentence “ብልፅግና ፓርቲ የገዳዮች ስብስብ ነው”. The tokenized result will be

['ብልፅግና', 'ፓርቲ',' የገዳዮች',' ስብስብ',' ነው'].

Character normalization in Amharic text aims to ensure consistency across characters. In the Amharic script, characters that produce the same sound may have distinct symbols, known as homophone characters. For instance, the characters 'ሠ' and 'ሰ' might be used interchangeably in words like "ሠው" and "ሰው", both meaning "man". This study addresses such inconsistencies in word writing by substituting characters that produce the same sound with a standardized form.

Lemmatization is the process of reducing a word to its lemma, which is its dictionary or base form a valid word in the dictionary. Unlike stemming, lemmatization considers the context and part of speech (POS) of the word to return a valid word. Lemmatization usually requires more advanced tools, such as a lexicon or morphological analysis, to handle inflections correctly.

Amharic words can take many forms due to morphological complexity: Example: "እየመጣናል", "መጣ", and "መጣች" share the root "መጣ" (come).

Stemming: Strip affixes to find the stem of the word: Example: "ተመልሷል" → "መልስ".

Lemmatization: Reduce words to their base dictionary form: Example: "እየተማርኩኝ" → "ማርከ".

6. Feature Extraction Techniques Using Word2vec

For the preparation of word2vec model we have collected 446214 corpora datasets extracted from Facebook, Twitter and YouTube social media. The parameter configuration of Word2Vec directly affects the quality of word vectors. Because small word embedding are not sufficient to express all possible word relationships, very large word embedding's leads to overfitting [18] [19].

In this thesis we used both CBOW and skip-gram in order to determine which model is better for performing the intended tasks. First, we made an experiment for word2vec with CBOW models with embedding dimensions 400 for each word in the dataset, and the minimum word count becomes 3 and with window size 5. This parameter was chosen after conducting different experiments by comparing the outcomes with other experiments.

7. Feature Extraction Techniques Using FastText

The same number of unlabeled corpora used for preparing FastText model like word2vec model as discussed previously. FastText solve out of vocabulary problem by inheriting relationship from its character n-grams even if word doesn't appear in training corpus [20]. This feature enhances learning on heavily inflected languages. Word2vec leaves unseen words as out-of-vocabulary words. So, with this intuition, we proposed to use FastText as word embedding techniques for generating better word vectors. FastText bring similar words in one vector space which is important for the identification of sentiment words contextually with spelling errors.

The CBOW and skip-gram architecture of FastText are used for identifying the better architecture by conducting different

experiment for each. Initially we used FastText with CBOW model with a fixed size of 200 for every word with minimum count of four words and a window size of five.

Again, we made the second experiment with skip-gram architecture by applying with the same number parameters like CBOW for extracting features from textual data. As a result, we got better result using this word representation techniques. Unlike word2vec for building Fast Text we used fixed size of 200. Because higher embedding dimension for FastText is very resource intensive, as it requires huge amount of RAM for storing the word vector of the corpus. Because FastText can treat each character as an atomic entity or vocabulary.

8. Deep learning approaches

Numerous deep learning models exist for sentiment classification. In this study, the selection of deep learning models was guided by their suitability for sentiment analysis in Amharic. The decision-making process considered criteria outlined[21][22] . These criteria encompassed attributes like efficient feature extraction, handling long-term dependencies, mitigating the vanishing gradient problem, interpreting diverse linguistic contexts, and employing models with fewer parameters to facilitate faster convergence.

Long Short-Term Memory (LSTM) networks, a distinctive variant of Recurrent Neural Networks (RNNs), excel at capturing long-term dependencies[23]. Fundamentally, LSTM retains information from previously processed inputs through a hidden state mechanism. Within the realm of LSTM networks, two primary configurations exist: unidirectional and bidirectional models.

Unidirectional LSTM focuses solely on past information as it processes inputs only from previous instances, whereas bidirectional LSTM operates in two directions: past to future and future to past. The cell state within LSTM plays a crucial role in retaining vital information throughout sequence processing [24]. During its progression, the cell state undergoes modifications, with information being either added or removed through specialized gates. LSTMs, by design, excel in long-term information retention within Recurrent Neural Networks (RNNs). This capability is attributed to the LSTM's memory structure, akin to a computer's memory. The LSTM's memory functions akin to a Gated cell, where the term "gated" refers to the cell's ability to decide whether to store or delete information (i.e., opening or closing gates) based on the information's assigned value [25][26].

Within LSTM architecture, three essential gates govern information flow: the input gate, the forget gate, and the output gate. These gates determine the inclusion of new input (input gate), the necessity of discarding unnecessary information (forget gate), and the impact of information on the current output (output gate) at a given timestamp. Operating as analog sigmoid functions within the range of 0 to 1, the gates within an LSTM network effectively address the issue of vanishing gradients. By maintaining gradients at sufficiently steep levels, LSTMs facilitate rapid training and

achieve high levels of accuracy. The Figure 2 provided below illustrates the graphical representation of an LSTM network layer.

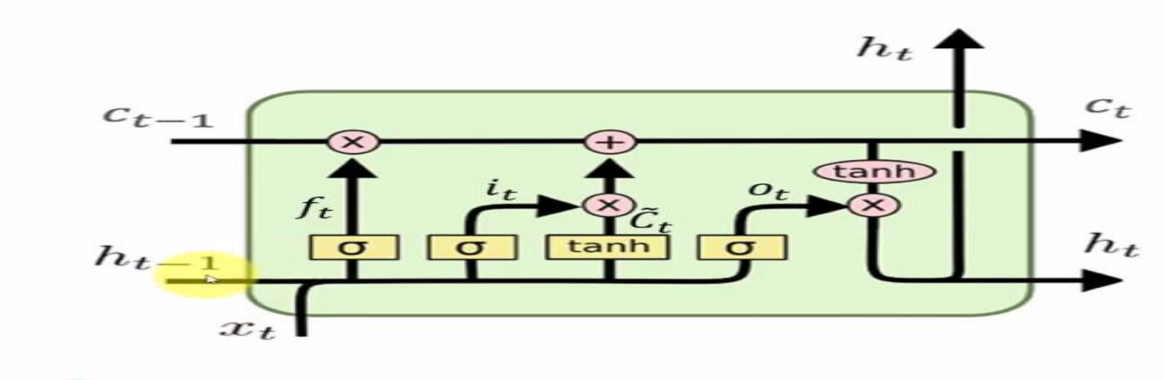

Figure 2. LSTM Network Layer

Bidirectional Recurrent Neural Network (BiLSTM):-The knowledge preserved by the unidirectional LSTM is limited to the past [27] . Bidirectional Long Short-Term Memory networks (BiLSTM) essentially combine two separate LSTM units. This arrangement enables the network to capture both forward and backward information regarding the sequence at each time step. By integrating forward and backward contexts, BiLSTMs expand the volume of available data for the network, providing richer context such as understanding the words that come before and after a specific word within a sentence[28] . The Bidirectional LSTM allows you to perform input in both directions, from the past to the future, and the other to the future past. The Figure 3 provided below illustrates the graphical representation of BILSTM network layer.

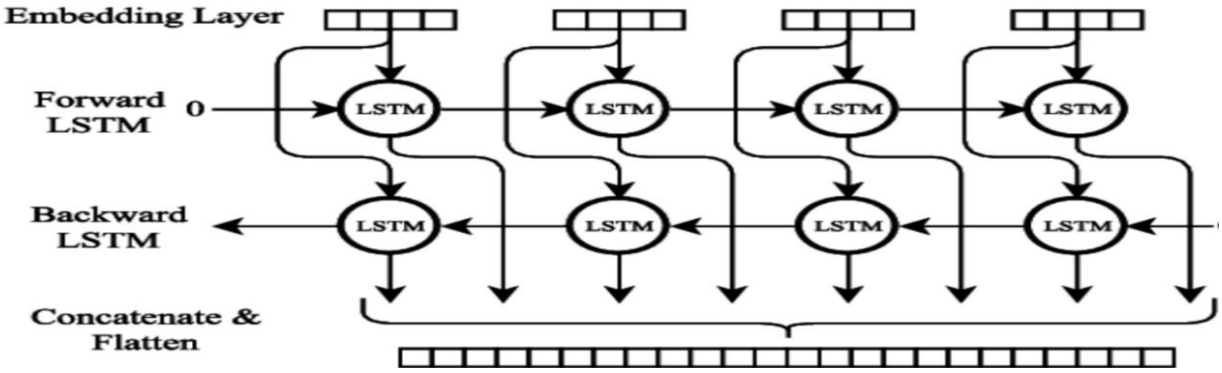

Figure 3. Bidirectional Long-Short Memory Networks

The Gated Recurrent Unit (GRU) is a type of recurrent neural network akin to LSTM but comprises solely two gates — the reset gate and the update gate — lacking an output gate [29]. GRUs are generally more straightforward to train than LSTMs due to

their reduced parameter count. While GRUs offer speed and memory efficiency advantages over LSTMs in certain scenarios [30], LSTMs excel when handling datasets with prolonged sequences. Given their similar structure and comparable performance in various contexts, GRUs can be viewed as a variant of the LSTM unit. By leveraging the update and reset gates, GRUs effectively combat the issue of vanishing gradients. The update gate manages incoming memory information, whereas the reset gate regulates the outgoing memory flow. GRUs exhibit superior performance to LSTMs, particularly with smaller datasets.

9. Experiment and Results

In this study, we conducted different experiments using different deep learning algorithms, such as LSTM, BiLSTM, and GRU, with different word embedding techniques, namely, word2vec and FastText. To perform these experiments, we used our prepared Amharic datasets, which were collected from social media in political domains. The labeled dataset contains 3753, 3588, and 2242 negative, positive, and neutral comments, for a total of 9583 comments. We have used the total corpus size of 446214 collected from social media in different domains to generate vocabulary using word2vec and FastText. After that, before feeding the prepared dataset to the deep learning algorithm for training, we should split the dataset into training and testing. To do this, we used the train_test_split module provided by the sickit-learn library. Therefore, in this thesis we have used approximately 90% of the dataset for training, 10% for testing. Since then, we have gotten better results using such a ratio. After that, we feed the input corpus into the input layer of the recurrent neural network. The parameters become tuned to deliver the right result for a given input. Finally, the model's predictions are evaluated on the test set. Results of the evaluation metrics using various deep learning models with Word2Vec word embedding's are outlined below.

Table 2. Evaluation Metrics Results with Different Deep Learning through Word2vec Word Embedding.

Model	Class	Precision	Recall	F1-Score	Accuracy
LSTM + Word2vec	Positive	0.82	0.81	0.81	0.82
	Negative	0.80	0.82	0.81	
	Neutral	0.85	0.84	0.85	
BILSTM + Word2vec	Positive	0.82	0.85	0.83	0.83
	Negative	0.82	0.85	0.84	
	Neutral	0.90	0.80	0.85	
GRU + Word2vec	Positive	0.84	0.74	0.79	0.81
	Negative	0.77	0.86	0.81	
	Neutral	0.86	0.83	0.85	

As we see in Table 2 above, we have conducted different experiments using different deep learning algorithms—LSTM, BiLSTM, and GRU—with wod2vec embedding as a feature extraction technique. As a result, we achieved an accuracy of 0.82% with the LSTM algorithm and 0.81% with the GRU, and we achieved a better result through BiLSTM, with an accuracy of 0.83%. The visualization of the classification report for a BiLSTM model with FastText embeddings is presented Figure *4*.

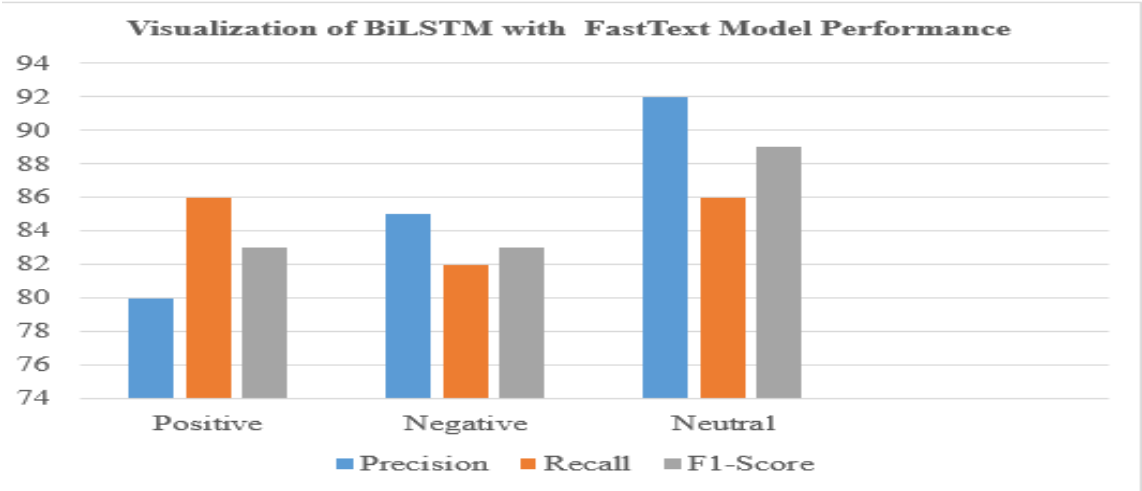

Figure 4. Visualization Classification Report for BiLSTM with FastText

Table 3: Evaluation Metrics Results with Different Deep Learning Algorithms through FastText Word Embedding.

Model	Class	Precision	Recall	F1-Score	Accuracy
LSTM + FastText	Positive	0.80	0.85	0.82	0.83
	Negative	0.84	0.81	0.82	
	Neutral	0.88	0.85	0.87	
BiLSTM + FastText	Positive	0.80	0.86	0.89	0.84
	Negative	0.80	0.86	0.83	
	Neutral	0.92	0.86	0.89	
GRU + FastText	Positive	0.83	0.79	0.81	0.83
	Negative	0.81	0.85	0.83	
	Neutral	0.87	0.87	0.87	

As we see in

above, we have conducted different experiments using different deep learning algorithms—LSTM, BiLSTM, and GRU—with FastText embedding. We achieved an accuracy of 0.83% with the LSTM algorithm and 0.83% with the GRU, and we achieved a better result through BiLSTM, with an accuracy of 0.84%.

Table 2 we performed the first experiments using LSTM, BILSTM, and GRU with word2vec as a feature extraction technique, and we obtained accuracies of 82%, 83% and 81% (LSTM, GRU and BILSTM), respectively.

As we observed from the experimental results, the BILSTM deep learning algorithm achieved better results with word2vec embedding techniques than did the LSTM and GRU deep learning

10. Discussion of the Results

In this study, we performed different experiments to distinguish the best performing models using different deep learning algorithms, namely, LSTM, GRU, and BiLSTM, with different word embedding techniques for feature extraction, namely, word2vec and FastText. As shown in

algorithms. This is due to the ability of BILSTM to learn forward and backward contextual information from text. As shown in

we performed the second experiment again using the LSTM, BiLSTM, and GRU deep learning algorithms by applying FastText as a feature extraction technique. As a result,

we obtained accuracies of 83%, 83%, and 84.25% (LSTM, GRU and BILSTM, respectively).

According to the experimental results from this research, the BILSTM deep learning algorithm with FastText as an embedding technique achieved better results with an accuracy of 84.25%. This is due to the ability of FastText to consider words in different ways, which means that FastText by default is a character n-gram. It may consider 1, 2, 3 or more grams as needed, and its ability to capture semantic information is better than that of word2vec. It handles words out of vocabulary, which means that even if words are not present in the vocabulary, they inherit some meaning from the available vocabulary.

In the developed model, the number of misclassified data points for the negative and positive classes is greater than that for the neutral class. This is due to the presence of some positive words in the sentence, but it may contain hidden meanings other than the common meaning (sarcasm content) that is classified as the negative class. In addition, sometimes there is also some text that contains negative words in the sentence, but it may contain hidden meaning that may be classified as positive.

For these reasons, the developed model is confused when incorporating essential features, and the occurrence of the same word in different contexts could cause incorrect features to be used by the models and incorrectly classify the comments. The other reason is that the imbalanced nature of the dataset also affects the performance of the model because, as we know about social media, the number of neutral comments is less than that of positive and negative comments. However, by using techniques such as oversampling and under sampling, imbalanced data can be handled, but these techniques are not effective, which leads to other problems such as overfitting and underfitting.

Generally, the accuracy of the classifier improves because the proposed model can handle sarcasm. However, due to the challenging nature of sarcasm expression, the proposed model may sometimes be confused when predicting the sentiment of a sarcasm expression. As a result, it is very difficult to develop a model that has better accuracy and can handle sarcasm expression effectively, such as normal expression.

11. Conclusions

In this research endeavor, we engineered a sentiment classification framework for Amharic social media text employing deep learning methodologies, accounting for sarcasm expressions prevalent in posts and comments, particularly within the political sphere where users commonly employ such expressions. Our investigative approach involved gathering Amharic content from social media platforms within the political landscape. Subsequently, crucial preprocessing procedures were implemented to refine our classification model. The dataset was meticulously annotated into three distinct categories—positive, negative, and neutral—by two proficient experts well-versed in Amharic language and political science.

Subsequently, we crafted the Word2vec and FastText word embedding models to produce word vectors capable of capturing both syntactic and semantic relationships among words, in addition to utilizing randomly generated vectors through the embedding layer. Following this, we executed an experiment employing state-of-the-art deep learning algorithms like LSTM, BiLSTM, and GRU, utilizing the FastText and Word2Vec embedding vectors. Through iterative parameter adjustments within the BiLSTM model, we attained enhanced performance, achieving an accuracy rate of 84.25% with FastText embeddings.

Nevertheless, this outcome might fall short compared to findings in other studies within the domain of natural language processing, particularly sentiment analysis. This discrepancy can be attributed to the intricate nature of sarcasm expressions, posing challenges even for humans to decipher due to their ambiguous nature. To date, no investigations have been undertaken concerning sentiment classification of Amharic text in the political realm involving sarcasm expressions. In essence, the developed model stands poised to aid political entities and voters in swiftly analyzing diverse viewpoints, thereby reducing the time and effort required to sift through numerous comments and posts.

Acknowledgments

This work was supported by the Debre Markos Institute of Technology, Debre Markos University.

Data availability

The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Competing interests

The authors declare no competing interests.

References

[1] N. S. A. Zulkifli and A. W. K. Lee, *Sentiment Analysis in Social Media Based on English Language Multilingual Processing Using Three Different Analysis Techniques*, vol. 1100, no. September 2019. Springer Singapore, 2019. doi: 10.1007/978-981-15-0399-3_30.

[2] Y. Getachew and A. Alemu, “Deep Learning Approach for Amharic Sentiment Analysis,” no. November 2018, 2019.

[3] S. Pedipina, S. Sankar, and R. Dhanalakshmi, “Sentimental Analysis on Twitter Data of Political Domain,” no. 07, pp. 205–216, 2021, doi: 10.1007/978-981-16-0965-7_17.

[4] P. Paper, H. D. Sharma, and P. Goyal, “An Analysis of Sentiment : Methods , Applications ,” no. Ml, 2023.

[5] F. Alemayehu, M. Meshesha, and J. Abate, “Amharic political sentiment analysis using deep learning approaches,” *Sci. Rep.*, vol. 13, no. 1, 2023, doi: 10.1038/s41598-023-45137-9.

[6] S. G. T. B and R. Damaševiˇ, “Deep Learning-Based Sentiment Classification,” vol. 1, pp. 63–75, 2022, doi: 10.1007/978-3-031-22792-9.

[7] B. Abebaw, “Sentiment Analysis of

Social Media Amharic Texts Using Word2vec Msc .," no. September, 2020.

[8] O. Kwon, D. Kim, S. R. Lee, J. Choi, and S. K. Lee, "Handling out-of-vocabulary problem in hangeul word embeddings," *EACL 2021 - 16th Conf. Eur. Chapter Assoc. Comput. Linguist. Proc. Conf.*, pp. 3213–3221, 2021.

[9] S. K. Alaramma, A. A. Habu, B. I. Ya'u, and A. G. Madaki, "Sentiment analysis of sarcasm detection in social media," *Gadau J. Pure Allied Sci.*, vol. 2, no. 1, pp. 76–82, 2023, doi: 10.54117/gjpas.v2i1.72.

[10] S. Hiai and K. Shimada, "Sarcasm detection using features based on indicator and roles," *Adv. Intell. Syst. Comput.*, vol. 700, no. February, pp. 418–428, 2018, doi: 10.1007/978-3-319-72550-5_40.

[11] P. Hantsch and N. Chkroun, "connotation_clashers at SemEval-2022 Task 6: The effect of sentiment analysis on sarcasm detection," *SemEval 2022 - 16th Int. Work. Semant. Eval. Proc. Work.*, pp. 945–950, 2022, doi: 10.18653/v1/2022.semeval-1.132.

[12] S. K. Bharti, K. S. Babu, and S. K. Jena, "Parsing-based sarcasm sentiment recognition in Twitter data," *Proc. 2015 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2015*, pp. 1373–1380, 2015, doi: 10.1145/2808797.2808910.

[13] R. Bose, R. K. Dey, S. Roy, and D. Sarddar, "Analyzing political sentiment using Twitter data," *Smart Innov. Syst. Technol.*, vol. 107, no. May, pp. 427–436, 2019, doi: 10.1007/978-981-13-1747-7_41.

[14] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "Improving sentiment analysis in Arabic using word representation," *arXiv*, no. February, 2018.

[15] M. Oljira, "Sentiment Analysis for Afaan Oromoo Using Combined Convolutional Neural Network and Bidirectional Long Short-Term Memory," vol. 11, no. 11, pp. 101–112, 2020, doi: 10.34218/IJARET.11.11.2020.010.

[16] S. M. Yimam, H. M. Alemayehu, A. Ayele, and C. Biemann, "Exploring Amharic Sentiment Analysis from Social Media Texts: Building Annotation Tools and Classification Models," pp. 1048–1060, 2021, doi: 10.18653/v1/2020.coling-main.91.

[17] Y. TESHOME, "Sentence Level Opinion Mining for Amharic Language," no. June, 2019.

[18] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, "Sentiment Analysis Using Word2vec and Long Short-Term Memory (LSTM) for Indonesian Hotel Reviews," *Procedia Comput. Sci.*, vol. 179, no. 2020, pp. 728–735, 2021, doi: 10.1016/j.procs.2021.01.061.

[19] I. A. Asqolani and E. B. Setiawan, "A Hybrid Deep Learning Approach Leveraging Word2Vec Feature Expansion for Cyberbullying Detection in Indonesian Twitter," *Ing. des Syst. d'Information*, vol. 28, no. 4, pp. 887–895, 2023, doi:

10.18280/isi.280410.

[20] B. A. Putri and E. B. Setiawan, "Topic Classification Using the Long Short-Term Memory (LSTM) Method with FastText Feature Expansion on Twitter," *2023 Int. Conf. Data Sci. Its Appl. ICoDSA 2023*, no. August 2023, pp. 18–23, 2023, doi: 10.1109/ICoDSA58501.2023.10277033.

[21] A. R. Andriawan, M. Mustakim, and R. Novita, "Sentiment Analysis Classification Of Political Parties On Twitter Using Gated Recurrent Unit Algorithm And Natural Language Processing," *J. Informatics Telecommun. Eng.*, vol. 7, no. 2, pp. 514–522, 2024, doi: 10.31289/jite.v7i2.10709.

[22] S. M. Yimam, H. M. Alemayehu, A. A. Ayele, and C. Biemann, "Exploring Amharic Sentiment Analysis from Social Media Texts: Building Annotation Tools and Classification Models," *COLING 2020 - 28th Int. Conf. Comput. Linguist. Proc. Conf.*, pp. 1048–1060, 2020, doi: 10.18653/v1/2020.coling-main.91.

[23] J. Nowak, A. Taspinar, and R. Scherer, "LSTM recurrent neural networks for short text and sentiment classification," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10246 LNAI, no. November, pp. 553–562, 2017, doi: 10.1007/978-3-319-59060-8_50.

[24] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *30th Int. Conf. Mach. Learn. ICML 2013*, no. PART 3, pp. 2347–2355, 2013.

[25] J. H. Wang, T. W. Liu, X. Luo, and L. Wang, "An LSTM approach to short text sentiment classification with word embeddings," *Proc. 30th Conf. Comput. Linguist. Speech Process. ROCLING 2018*, pp. 214–223, 2018.

[26] A. Tholusuri, M. Anumala, B. Malapolu, and G. Jaya Lakshmi, "Sentiment analysis using LSTM," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6 Special Issue 3, pp. 1338–1340, 2019, doi: 10.35940/ijeat.F1235.0986S319.

[27] Z. Hameed and B. Garcia-Zapirain, "Sentiment Classification Using a Single-Layered BiLSTM Model," *IEEE Access*, vol. 8, pp. 73992–74001, 2020, doi: 10.1109/ACCESS.2020.2988550.

[28] B. Iung, "Cœur et grossesse," *EMC - Trait. médecine AKOS*, vol. 8, no. 2, pp. 1–4, 2013, doi: 10.1016/s1634-6939(13)59289-1.

[29] Y. Gao and D. Glowacka, "Deep gate recurrent neural network," *J. Mach. Learn. Res.*, vol. 63, pp. 350–365, 2016.

[30] Y. Santur, "Sentiment analysis based on gated recurrent unit," *2019 Int. Conf. Artif. Intell. Data Process. Symp. IDAP 2019*, pp. 1–5, 2019, doi: 10.1109/IDAP.2019.8875985.