Bilingual Social Media Text Hate Speech Detection For Afaan Oromo And Amharic Languages Using Deep Learning

Baharudin Sherif Kemal[1], Teklu Urgessa Abebe[2], G.V.S.Kumar Pendem[3], T.Gopi Krishna[4], Ketema Adere Gemeda[5]

[1]Department of Computer Science, College of Engineering and Technology, Mattu University, Mattu, Ethiopia.
[2,3,4,5]Department of Computer Science and Engineering, School of Electrical Engineering and Computing, Adama Science and Technology University, Adama, Ethiopia.

Abstract

In Ethiopia, the problem of hate speech posts on social media has become challenging recently. Detecting hate speech posts on social media is a tedious and complex task due to the unstructured format of social media content, which requires some detection mechanisms. Due to the success of deep learning algorithms in natural language processing tasks, some researchers used deep learning models for hate speech detection. But, most of the existing studies are explored only for high resource languages like English, except some studies recently proposed also for low resource languages. This study proposedbilingual hate speech detection for Afaan Oromo and Amharic texts on social media using deep learning. A bilingual dataset prepared from newly collected Afaan Oromo texts from Facebook platform and the existing binary Amharic dataset is adopted to develop models. The prepared dataset contains binary classes “Hate” and “Free”. Bidirectional RNNs and attention mechanisms are implemented using Word2vec as feature representation. The word2vec model is trained based on the skip-gram model. The models are trained using 5-fold and also 10-fold cross-validation. The results show that, models achieved a good performance when using 5-fold cross-validation on our dataset. Then, several experiments are employed to select the best-performing model, and finally, the BiLSTM model outperformed all other models with an accuracy of 94.3% and f1_score of 94.2%.

I. INTRODUCTION

Recently, the user of social media has grown significantly in Ethiopia. Social media platforms can disseminate information quickly and widely as a means of communication. It has opened up a new galaxy of opportunities for people to express their opinions freely. Nowadays people are using social media for online trading, dissemination of government policies, political campaigns, and religious preaching. Despite its importance, social media created an opportunity for the propagationof hate speech and disinformation online[1]. Having many audiences, some peoples use to share their hate speech propaganda online. Online hate speech can be accessed by a large number of the audience quickly which makes the problem very difficult. The term hate speech refers to a speech that promotes violence or attack on any protected category of people based on their race, religion, sex, ethnicity, national origin, sexual orientation, gender, or disability[2]. It includes all kinds of expressions that promote attack and discrimination or incite violence against a targeted group of people or individuals based on their protected identity. Hate speech towards targeted groups or individuals can lead to serious conflicts on the ground by motivating people for criminal actions. In Ethiopia,there is a diversity of cultures, religions, ethnic groups,and languages. Any kind of hate speech against a targeted group or individuals can be a reason for conflict in Ethiopia. Hence, Ethiopia posted a hate speech prevention proclamation recently.It defined the term hate speech as'speech that deliberately promotes hatred, discrimination or attack against a person or a discernable group of identity, based on race, ethnicity, religion, gender or disability.The USA grants companies of social media broad managing power of their content and enforcing the rule of hate speech. Companies in different countries like Germany were forced by the government to remove posts within certain periods. In Ethiopia, the result of hate speech posts on social media by different individuals made several conflicts[3].

Social media platforms includingFacebook, Instagram, and Twitterare struggling to use artificial intelligence (AI) applications on their site to block hate speech automatically and make safe environments for their customers.In the past few years, researchers were focusing on the problem of hate speech detection and proposed different NLP and machine learning techniques to solve the problem. Most existing works on the detection of hate speech are explored for some high-resource languages like English. But nowadays people are using social media as a playground for hate speech and insults against targeted groups or individuals. Hence, there should be a mechanism for hate speech detection also for under-resourced languages. There are also some other existing works on hate speech detection using machine learning and deep learning for several languages[4], [5]. However, there is no research conducted on bilingual hate speech detection for Afaan Oromo and Amharic languages. This research work proposedbilingual social media hate speech detection for Afaan Oromo and Amharic languages. The main goals of this research are to prepare bilingual Afaan Oromo and Amharic dataset, and to develop deep learning models for classifying Afaan Oromo and Amharic language texts as hate or free using bilingual data.Major contributions of the studyinclude:

- ✓ Building bilingual hate speech dataset and applying bidirectional recurrent neural networks.
- ✓ Developing baseline attention-based deep learning architectures for detection of Afaan Oromo and Amharic hate speech texts.
- ✓ Development of the hate speech detector prototype for reading Afaan Oromo and Amharic language posts and comments of specific users and classify whether it is hate speech or not.

Further, this work includesevaluation of different deep learning models on bilingual data for the detection of social media hate speech.Hyperparameter tuning is also another operation included in the experiment to select proper combination of hyperparameters for the improvement of hate speech detection mechanism.

II. LITERATURE SURVEY

Hate speech is a crime that has been growing especially in online communications. The internet, social media platforms, and the increased willingness of people to express their opinions online contributing to the propagation of hate speech very fast[6].Since the problem of social media texts hate speech detection is related to text classification tasks, many researchers applied machine learning and deep learning techniques for the problem of online hate speech detection.

The work by[7]proposed a hate speech detection model for Amharic language using machine learning and text-mining feature extraction techniques. They collected data from selected Facebook public pages and manually labeled them into three classes and then converted them into binary classes to build binary and ternary datasets. SVM, NB, and RF models trained using the whole dataset with the extracted feature based on word unigram, bigram, trigram, combined n-grams, TF-IDF, combined n-grams weighted by TF-IDF and word2vec for both datasets. The models were evaluated using 5-fold cross-validation. The models based on SVM with word2vec perform better in classification results with a 73% F1-score, and it shows slightly better performance than the NB and RF models for both binary and ternary models. However, the researchers were limited to using only machine learning algorithms and they developed the models only for the Amharic language. Deep learning models can be implemented for better classification performance.

[8] proposed Amharic text hate speech detection using deep learning approaches. The author collected data from Facebook and Twitter and labeled them into four classes as hate, offensive, both (hate and offensive), and neutral. Word embedding using Keras and Word2Vec embedding used as feature extraction. For the experiment LSTM, CNN, BiLSTM, combined CNN-LSTM, and GRU models were trained using the whole dataset. The researcher used (80,20) train-test split with performance metrics of precision, recall, and

f1-score for model comparison. The researcher applied data augmentation techniques and formed two datasets (original and augmented). Five different models were developed using both datasets and the model based on BILSTM with word2vec achieved better performance than the other models for both augmented and original datasets with an accuracy of 88.89% and an f1-score of 89% for the original dataset.The authors are limited to only using RNNs, BiRNNs, and CNN models, while attention mechanisms can be implemented, also they didn't use cross-validation technique which helps to minimize the problem of overfitting. They developed the models only for the Amharic language.

[9]presenteda mechanism for developing a modelfor the detection of hate speech and offensive language on Twitter by using Naive Bayes, Logistic Regression, and Support Vector Machines with n-gram features weighted with TF-IDF values using a publicly available Twitter dataset. The authors used a very large number of datasets combined from three different publicly available datasets from Crowd flower containing tweets that have been manually classified into Hateful, Offensive, and Clean. Logistic Regression outperformed other models with the optimal n-gram range of 1 to 3 for the L2 normalization of TF-IDF and the model achieved 95.6% accuracy. The authors implemented only machine learning models for monolingual data, but our study includes evaluation of deep learning models for bilingual data.

The work presented by[10] developed models to detect hate speech in the Indonesian Language from input text and speech by using a deep learning approach. The authors used both textual and acoustic features and compare their accuracies. From the experiment, better results for hate speech detectionare obtained using only textual features is better than that of using acoustic features and both combined features models. The best model using textual feature obtained an F1-score of 87.98% which is higher than the model of using acoustic feature only (F1-score 82.5%) and the model of using acoustic and lexical features (F1-score 86.98%). Although the researchers tried to use deep learning models, our study aimed to consider bilingual data and also, to achieve better performance for hate speech detection.

[11] Developed recurrent neural network models for automated hate speech post-detection for Amharic posts on Facebook. The long short-term memory and gated recurrent unit with word n-grams for feature extraction were used by the authors. They used word2vec to represent each unique word by vector representation.The authors split the dataset intoa train set, validation set, and test set of 80%, 10%, 10% respectively for the experiment. LSTM based RNN with Batch size 128, learning rate 0.001 with 0.5 dropouts, and RMSProp optimizer achieved an accuracy of 97.9% to classify posts as free or hate speech by training with 100 epochs. Despite they achieved good accuracy, they are limited to develop a model for a monolingual dataset. further, other

deep learning models such as attention mechanism and bidirectional recurrent neural networks can be tested instead of using only LSTM and GRU.

[12]proposed an Italian online hate campaign on the social network. They used data collected from Italian public Facebook pages. The dataset is annotated into three classes as no hate, weak hate, and strong hate, and by merging weak and strong hate as hate they form the second dataset. The author design and implement two classifier algorithms for the Italian language, SVM and LSTM algorithm leveraging morpho-syntactical features, sentiment polarity, and word embedding lexicons. Conducting two different experiments with both datasets that a least 70% of the annotator agreed on the class of the data. F1-score of 80% is achieved by using SVM and LSTM achieved an f1-score of 79% for binary classification. For the ternary classification SVM and LSTM achieved f1_score of 64% and 60% respectively. They used only the conventional SVM and RNN model LSTM, while more deep learning models can be implemented to improve the classification performance.

[13]Proposed Cyberhate speech detection based on Arabic context over the Twitter platform, by applying NLP and machine learning techniques. The work focused on a set of tweets related to sport, racism, terrorism, journalism, sports orientation, and Islam. The processed dataset is experimented with using Decision Tree (DT), Support Vector Machine (SVM), Naive Bayes (NB), and Random Forest (RF). In their experiment, RF with TF-IDF and profile-related features achieved a better result with an accuracy of 91.3%. Since hate as a term is subjective and can be expressed in a wide range of areas not restricted to the sport, religious or racial issues they recommended further work for the more generalized dataset and effective detection models. The authors recommended implementation of effective detection models. In Our study, we tried to apply different deep learning models to achieve better performance.

Likewise, the deep learning approach for automatic cyber hate speech detection on Twitter is presented by [14]. The dataset for the study was collected from Twitter and the collected data captures different hate expressions in the Arabic region. The author’s used word embedding mechanisms for feature extraction. The hybrid of CNN and LSTM network is implemented for model development. The proposed approach aimed to classify tweets as hate and normal and achieved promising results, 66.564%,79.768% 65.094%,71.688% regarding the accuracy, recall, precision, and F1 measure respectively. Thestudy recommended a more standardized dataset and high-performance deep learning approaches.

Another work by [15] proposed comparative analysis of deep learning models for Afaan Oromo hate speech detection. They implemented different deep learning models including CNN, BiLSTM, LSTM and GRU. The authors collected total of 35,200datasets from selected Facebook pages and Twitter, and augmented it in to 42,100. Then, they labeled the dataset into four classes (hate, offensive, neutral, and both). After implementing models using the augmented dataset, BiLSTM model performed better with F1-score of 91%.

Table 1.Comparison of related works

Authors and Year	Data source, size, and classes	Method used and Performance achieved
Y. Kenenisa and T. Melak (2019)	Facebook: 5,000, They experimented with both binary and ternary class [Hate, Normal] & [Hate, Offensive, Neither]	SVM with word2vec F1-score: 73% (for binary) F1-score:53% (for ternary class)
F. Del Vigna, A. Cimino, F. Dell'Orletta, M. Petrocchi, and M. Tesconi (2017)	Facebook: 3,575 [no hate, weak-hate, and strong-hate]	LSTM with word embedding F1-score: 75.2%
I. Aljarah et al (2020)	Twitter: 3696 [Hate, not-hate]	RF with TF-IDF Accuracy: 91.3%
E. Baweke (2020)	Facebook and Twitter: 27495 [Hate, offensive, both and neither]	BiLSTM with word2vec Accuracy:88.89% f1-score:89%
H. Faris, I. Aljarah, M. Habib, and P. A. Castillo (2020)	Twitter: 3696 [Hate, normal]	LSTM- CNN Accuracy:66.564%, F1-score:71.688%
S. G. Tesfaye and K. K. Tune (2020)	Facebook: 30,000 [Hate, free]	LSTM Accuracy: 97.9%
T. L. Sutejo and D. P. Lestari (2018)	Facebook, Twitter, YouTube, and Line Today Audio:2469 Text: 2273 [Hate, no-hate]	LSTM: using textual features F1-score 87.98%
G. O. Ganfure (2022)	Facebook and Twitter: 35,200 (Augmented to 42,100) [Hate, Neutral, Offensive, Both]	BiLSTM Using Word2vec (CBOW) F1-score 91%

As presented in the **Table 1**, people tried different techniques for the task of hate speech detection. The problem is a supervised learning task as we need a labeled hate speech dataset for classification. The lack of a publicly available dataset is one of challenges limiting researchers to focus on a limited number of languages only. Eventhough people proposed several techniques for the problem of hate speech detection, we found an open issue including: No bilingual text hate speech detection is implemented for Afaan Oromo and Amhariclanguages, there is no publicly available bilingual Afaan Oromo and Amharic languages dataset, Also, no researchers used attention mechanism for the hate speech detection of Amharic and Afaan Oromo languages. Our study focused on those open issues.

Table 2.Sample Facebook posts of different classes

Label	Facebook posts	Language
Free	Dhugaa hin haalamneedha Ijaarsa biyyaa galma barbaadamuun gahuuf kopha kophaa fiiguurra humna misoomaa fi jijjiiramaa abdii biyyaa kan tae humna kanaaf xiyyeeffannoon kennamuu qaba	Afaan Oromo
Hate	Itiyoophiyaan oromoyaaf hin taatu egaa ta'ee wal haa fixxu Achii booda abbaan hafe haa jiraatuu itiyoophiyaan loliinsaan malee wolirraa hin dhaabattu	Afaan Oromo
Free	እንዴትደስስይላልእግዚአብሄርሁሉንምያሳካልን	Amharic
Hate	ያንተድንቁርናደግሞየምጀምረውደግሞገናለማንበብእንደጀመርኩኝቁጥርላይአድስአበባእንደገናከተቆረቆረችአመቷነውእዝህውስጥአንተናየአንተአይነቱየሰውነገርሁሉየምያምረውእንደገናየምትለዋቃላትለምንአስፈለገከዚያበፍትፍንፍነየምትባልአለመኖሯንነውየምያሳየውለላስምካወጣንላትጀምሮነውእድመዋየምቆጠረውነውካላልክበስተቀረመልሱንእራስህምመልሰሀል	Amharic

III. PROPOSED METHODOLOGY

A) Dataset

The dataset used for this study contains both Afaan Oromo and Amharic languages.For Afaan OromoLanguage the dataset is prepared from scratch by scraping posts and comments from the Facebook platform. After that,it is merged with the Amharic dataset and prepared the bilingual dataset for this study which contains both Afaan Oromo and Amharic languages. To preparethe Afaan Oromo hate speech datasetwe followed the following steps:

- ☞ Facebook pages selectionand scraping pages.
- ☞ Preprocessing the data and consolidating it into one file dataset.

☞ Annotating the data.

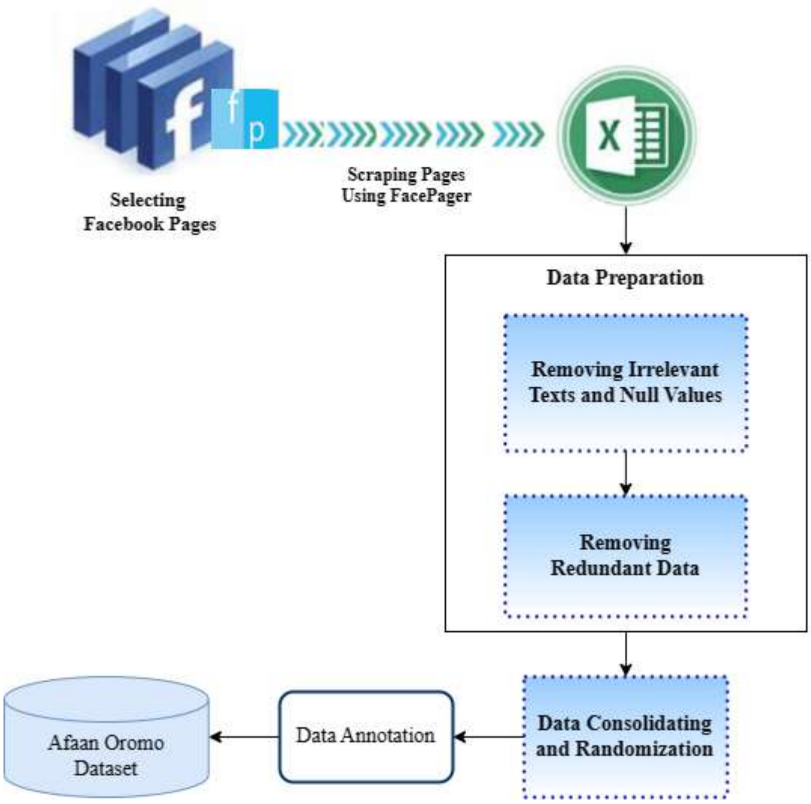

Selecting Facebook Pages
Scraping Pages Using FacePager
Data Preparation
Removing Irrelevant Texts and Null Values
Removing Redundant Data
Data Consolidating and Randomization
Data Annotation
Afaan Oromo Dataset

Fig.1. Strategy for collecting Afaan Oromo data

The posts and comments were gathered from different categories of a popular public page because Facebook's privacy policy does not allow access to the public content of a private page.Because of a lack of resources needed to manage and crawl all public Facebook pages in selected categories, we used only a limited number of Facebook pages. To manage the size of the data,we collected only posts and comments posted from September 2020 until May 2021.We used a Facepager which is Facebook Graph API-based data scraping tool. It is an HTTP-based API used to programmatically query data and performs a wide variety of other tasks.To have a representative dataset we identified criteria for Facebook page selection.The collected data is annotated based on an anti-hate speech guideline thatis prepared by the Center for Advancement of Rights and Democracy. In addition to the guideline, the CARD organization provided us a list of selected hateful keywords that we used during the data annotation process. The Amharic language dataset used for this study is contributed by other researcher, which is prepared by collecting posts and comments from the Facebook platform and published on Mendeley. It is available at DOI: 10.17632/ymtmxx385m.1.

B) Criteria for Facebook Page Selection

☞ A page that mostly uses the Afaan Oromo language for posts.

☞ Pages having likes and followers greater than 20,000.

- ☞ Pages of religious media, famous vlogers, politicians and broadcasting media are selected to get many and more representative data.

C) Data Preparation

Filtering and cleaning the data primarily use for the next stage, which is the annotation of the posts and comments in the dataset, and then used for training models. The following tasks were performed to prepare the dataset for annotation:

- ☞ Removing non-textual posts and comments.
- ☞ Removing irrelevant characters.
- ☞ Removing null, blank values, and extra whitespace.
- ☞ Combining the data into a single file.
- ☞ Removing duplication to ensure the uniqueness of each text in a dataset. Based on the prepared criteria, the selected pages, page categories, and the number of data collected from each page are given in**Table 3**.

Table 3. Sources of data and collected datasets

No	Page Name	Page Catagories	Followers	Likes	Collected Data
1	Oromia Media Network	Media/News Company	1,705,780	1,385,961	1,379
2	Ortodoksii page	Religious Organization	29,294	28,876	196
3	Oromia Broadcasting Service - OBS	Broadcasting & Media Production Company	259,218	243,762	2,878
4	Tvislaamaa	Broadcasting & Media Production Company	209,030	138,812	187
5	OBN Afaan Oromoo	Media/News Company	710,698	565,487	4,396
6	FBC Afaan Oromoo	Media/News Company	573,280	529,342	6,262
7	Raayyaa Abbaamacca	Interest	615,652	539,219	742
8	VOA Afaan Oromoo	Broadcasting & Media Production Company	901,579	795,725	320
9	Ahmedin Jebel official	Personal Blog	705,848	636,905	2,334
10	AndualemBafakadu Demelce 1	Personal Blog	306,679	293,425	4,092
11	Kello Media	News & Media Website	273,999	182,038	496
12	Hawi – Anole	Personal Blog	153,625	147,166	4,889
13	BBC News Afaan Oromoo	Media/News Company	786,299	644,027	2,515
14	Taye Dendea Aredo	Politician	412,812	381,139	2,973
15	Ustaz Kamil Shamsu	Public Figure	290,541	258,343	595
16	Addisu Arega Kitessa	Politician	170,474	156,555	2,604

17	Mana Lubummaa Oromiyaa	Religious Organization	55,236	51,231	3,690
18	ODP Official	Personal Blog	79,734	76,788	4,088
Total Number of DataFiltered					44,616
Total Number of Unique DataFiltered After Removing Redundancy					40,000

D) Data Annotation

The dataset is labeled into different classes in data annotation process. The data annotation task is time taking, and it needs budgets needed for annotators.we did not annotate all the collected data because of our limited time and budget,rather we used a random selection of texts to be annotated.Fiveannotators participated in the annotation task, four MSc students and one Ph.D. student. We selected them based on their willingness to perform the task and their background inthe Afaan Oromo language, they can read, write and understand the language.The annotation task is performed based on the annotation guideline prepared by Center for Advancement of Rights and Democracy (CARD) organization, which is available at www.cardeth.org.Based on this guideline, the annotators were instructed to label each post and comment to two different classes ‘hate’ and ‘free’. All annotators were instructed to annotate 52,00 instances of datasets among that 3700 instances are unique and 1500 instances are the same for all annotators. We used the same 1500 instances annotated by all annotatorsto measure the inter-rater agreement. The inter-rater agreement gave a kappa value of 0.664 and which is a good level of agreement.

The annotation results by each annotator are given in Table 4 and Table 5.

Table 4.Unique instances annotated by each annotator

Labels	Annotator1	Annotator2	Annotator3	Annotator4	Annotator5	Total
Free	1836	1846	1848	1,860	1,892	9,282
Hate	1864	1854	1852	1,840	1,808	9,218
Total	3,700	3,700	3,700	3,700	3,700	18,500

Table 5.Common instances annotated by each annotator

Labels	Annotator1	Annotator2	Annotator3	Ammotator4	Annotator5	Total by Voting
Free	685	700	709	706	701	703
Hate	815	800	791	794	799	797
Total	1,500	1,500	1,500	1,500	1,500	1,500

From annotation results, a total of 20,000 instances are annotated where 9,985 of them are labeled as “Free” and the rest 10,015 are labeled as “Hate” for the Afaan Oromo language. Finally, the bilingual dataset is prepared by merging the Afaan Oromo dataset with the Amharic language dataset. Since the Amharic language dataset contains 30,000 instances and the prepared

Afaan Oromo dataset contains 20,000 instances, totally we got a dataset with 50,000 instances. From the 50,000 instances of the bilingual dataset 24,036 of them are labeled as “Free” and the rest are labeled as “Hate”. The distribution of the dataset by class is given in **Table *6***.

Table 6.Dataset distribution by class

Labels	Total Number of Instances
Free	24,036
Hate	25,964
Total	50,000

IV. PROPOSED ARCHITECTURE

The proposed bilingualhate speech detection is aimed to classify Afaan Oromo and Amharic posts and comments on social media as hate, and free. As shown inFig.2,the proposed architecture contains components of preprocessing, feature representation, model building, and model evaluation. It takes the bilingual dataset containing Afaan Oromo and Amhariclanguages and the preprocessing techniques are applied to the dataset.

After preprocessing word embedding is used for representing the unique extracted words obtained from tokenization as a feature vector. Bidirectional recurrent neural networks (Bi-RNN), and attention-mechanism are used for model development. To select the best detection model, the models were evaluated using 5-fold cross-validation. The best performing model is selected for classifying the data into the classes of “hate” or “free” based on the evaluation results. Finally, the selected best-performing model is used to develop a prototype of the detection model that can take a new Afaan Oromo and Amharic texts as input and classifies the input as free, or hate speech.

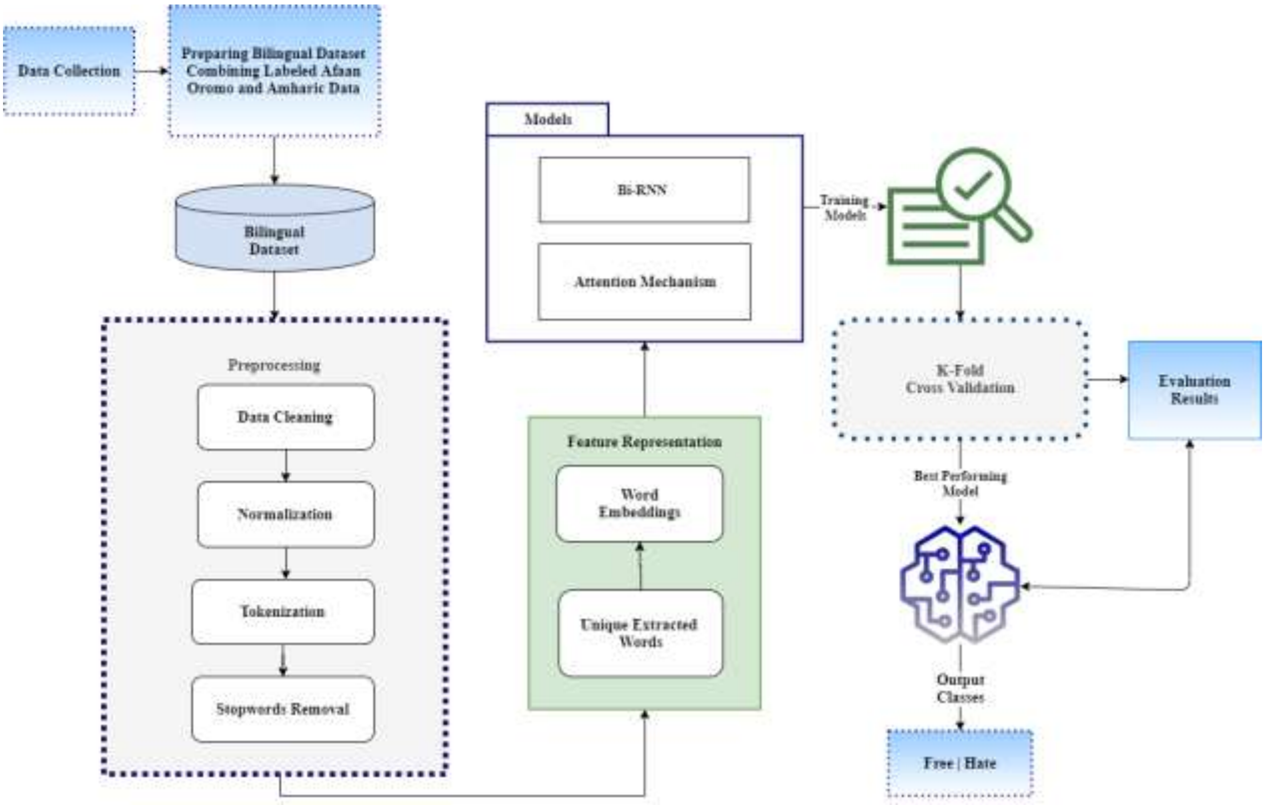


Fig.2. General Architecture for The Proposed Bilingual Hate Speech Detection Modeling

A) Preprocessing

Why preprocessing? Because the real-world data contains different unwanted contents, inconsistency and format that is difficult for the machine to analyze it. Therefore, we need to represent the data in such a way that it can be analyzed by machine, and this is what preprocessing steps do. The preprocessing step concerned with removing noises from the data or cleaning, normalization, and tokenization. As social media texts contain several less useful contents such as links, punctuations, and other special characters those should be removed during preprocessing for effective feature representation. Under this component preprocessing of Afaan Oromo and Amharic text is performed. This preprocessing component includes cleaning, normalization, and tokenization.

B) Cleaning

To remove unnecessary content and make the data more representable by the word embedding to be employed we shall go over some data cleaning steps. This cleaning procedure will get rid of all irrelevant special characters, symbols, and emojis. Pseudocode to remove irrelevant characters (cleaning) is as given below:

Algorithm 1 Data Cleaning Algorithm

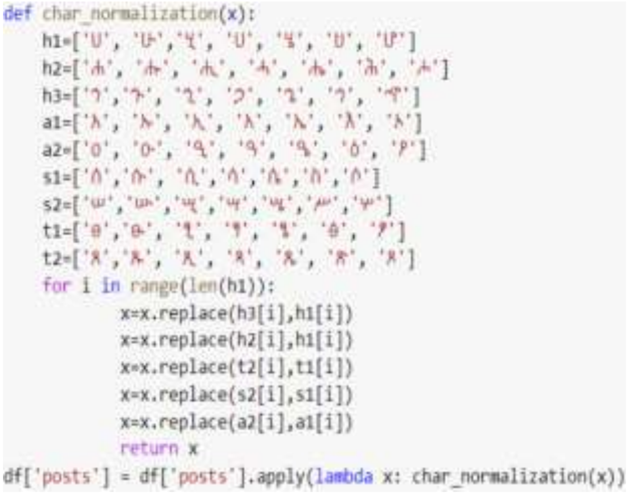
```
BEGIN:
1. Read the text in the dataset;
2. While (! end of the text in a dataset):
 If the text contains specialcharacters and symbols[',', '.', '"', ':', ')', '(', '-', '!', '?', '|', ';', "'", '$',  '&', '/', '[', ']', '>', '%', '=', '#', '*', '+', '\\', '•', '~', '@', '£', '·', '_', '{', '}', '©', '^', '®', '`', '<', '→', '°', '€', '™', '›', '♥', '←', '×', '§', '″', '′', 'Â', '', '½'] then
 Remove specialcharacters and symbols
 If the text contains number [0-9] &[ [፩፪፫፬፭፮፯፰፱፲፳፴፵፶፷፸፹፺፻] then
  Remove Arabic and Ethiopic numbers
  If a text contains emojis [😃, 😃, 🙈, 😛 ...] then
 Remove emojis
  If a text contains extra white space, then
  Remove extra space
3.Return clean_text;
END:
```

C) Normalization

Why normalization? Normalizing text is attempting to reduce its randomness and bring it closer to a predefined standard. Itis very useful to reduce the amount of different information the computer has to deal with and therefore improves efficiency. Its main goal is to group related tokens, where tokens are usually the words in the text.The task of text normalization includes converting texts to a similar case for the Afaan Oromo language. It is best to convert characters into lowercase since most of the time the users use lower case without dealing with the capitalization. Hence, all characters changed to lower case. For example, the word “UMMATA” is changed to “ummata” after changing to lower.Handling extra whitespaces is important to group the related tokens, therefore we applied those operations.For the Amharic language,the morphology is more complex and needs more normalization techniques. SomeAmharic letters have the same sound but are spelled differently. Therefore, people can write the same thing with a different spelling.For Example, በሌተዋሌand በሌቶኣሌ could be normalized to በሌቷሌ.We applied normalization on such Amharic letters.

Algorithm 2: Algorithm for Normalization

```
INPUT:Unprocessed dataset
OUTPUT: Normalized dataset
BEGIN:
1: Read the dataset
2: WHILE(it is not the end of file):
IF text contains [ሃኀኃሐሓኻ]' THEN replace with 'ሀ'
IF text contains [ሑኁዅ] THEN replace with 'ሁ'
IF text contains [ኂሒኺ] THEN replace with 'ሂ'
IF text contains [ኄሔዄ] THEN replace with 'ሄ'
IF text contains [ሕኅ] THEN replace with 'ህ'
IF text contains [ኆሖኾ] THEN replace with 'ሆ'
IF text contains [ዓኣዐ] THEN replace with 'አ'
IF text contains[ሠሡሢሣሤሥሦ]THEN replace with[ሰሱሲሳሴ ስ]
IF text contains [ዑዒዓዔዕዖኣ] THEN replace with [ኡኢአኤእኦአ]
IF text contains [ጸጹጺጻጼጽጾ] THEN replace with [ፀፁፂፃፄፅፆ]
IF text contains [ሉ[ዋአ]] THEN replace with [ሏ]
IF text contains [ሙ[ዋአ]] THEN replace with [ሟ]
IF text contains [ቱ[ዋአ]] THEN replace with [ቷ]
IF text contains [ሩ[ዋአ]] THEN replace with [ሯ] . . .
Return normalized text
ENDIF
3 END:
```



```python
def char_normalization(x):
    h1=['ሀ', 'ሁ','ሂ', 'ሀ', 'ሄ', 'ህ', 'ሆ']
    h2=['ሐ', 'ሑ', 'ሒ', 'ሓ', 'ሔ', 'ሕ', 'ሖ']
    h3=['ኀ','ኁ', 'ኂ', 'ኃ', 'ኄ', 'ኅ', 'ኆ']
    a1=['አ', 'ኡ', 'ኢ', 'ኣ', 'ኤ', 'እ', 'ኦ']
    a2=['ዐ', 'ዑ', 'ዒ', 'ዓ', 'ዔ', 'ዕ', 'ዖ']
    s1=['ሰ','ሱ', 'ሲ','ሳ','ሴ','ስ','ሶ']
    s2=['ሠ','ሡ','ሢ','ሣ','ሤ','ሥ','ሦ']
    t1=['ፀ','ፁ', 'ፂ', 'ፃ', 'ፄ', 'ፅ', 'ፆ']
    t2=['ጸ','ጹ', 'ጺ', 'ጻ', 'ጼ', 'ጽ', 'ጾ']
    for i in range(len(h1)):
            x=x.replace(h3[i],h1[i])
            x=x.replace(h2[i],h1[i])
            x=x.replace(t2[i],t1[i])
            x=x.replace(s2[i],s1[i])
            x=x.replace(a2[i],a1[i])
            return x
df['posts'] = df['posts'].apply(lambda x: char_normalization(x))
```

Fig.3. Coding Snippet for Amharic Letters normalization

D). Tokenization

Why tokenization? Tokenization is the process of converting text into tokens so that they can be considered as discrete values before transforming it into vectors. It is also easier to deal with and filter out unnecessary tokens. For example, a document into paragraphs or sentences into words.It detects the boundaries of a written text. In this step, input texts are tokenized into a stream of characters using white spaces and punctuation which helps to convert into a list of words. The process detects the boundaries of a written text. The input to deep learning models should be tokenized as text T = {$t1$, … tN}, where each word is represented by the word embeddings. For example: - The word “walqixxummaa sabaaf sablammii hundaa” tokenized as [“walqixxummaa”, “sabaaf”,”sablammii”, “hundaa”], and [“እርግጠኛ መሆን ኣለብህ”] tokenized as [“እርግጠኛ”,”መሆን”,”ኣለብህ”].

E). Stopwords Removal

Stop words removal is also another crucial step in preprocessing. There are many stopwords in Amharic and also Afaan Oromo language that is redundant in the dataset. Removing such stopwords is advantageous since we can get the core idea of the text even if they are removed, and it minimizes redundancy in the dataset which could improve classification performance. For example, in Afaan Oromo words like “akka”, “irra”, “hin” are redundantly appeared in the dataset, and removing those texts cannot change the idea of the text. In Amharic also words like “እኔ”, “እኛ”, “ነህ” are mostly redundant and less informative in the dataset. hence, we applied this preprocessing step for both languages Amharic and Afaan Oromo.

F). Feature Representation

The proposed feature representation component converts the dataset to feature vectors. As computers can’t understand text, feature representation techniques are used to change the text data to numerical form that could be understood by a machine. We used word embeddings for feature representation. Word embeddings are unsupervised learning of word representation whose relative similarity correlates with semantic similarity[16]. Hence, we used word embedding to get the advantage that they can model semantic similarity of words. We trained word2vec for the proposed bilingualhate speech detection by using posts and comments in the dataset without labels. Word2vec can be trained whether using skip-gram or continuous bag-of-words. For this study, the skip-gram model is used to create our word2vec model, in which the neural network finds the context words given the target word.

G) Models

Bidirectional RNN Models

We implemented RNN models for our proposed hate speech detection because of the suitability of RNN models for text data and the performance achieved by RNN models for text as presented by different literatures.

Bidirectional recurrent neural networks contain a combination of two recurrent neural network layers the backward and forward layers. The RNN added in the forward layer is used to read the input sequence beginning from start to end of the sequence and stores forward information (past) only. Then, the backward RNN layer is added to read the input sequence in reverse order from the end of the input sequence back to its beginning and store future information. After that, the combination of the forward information and the backward information is used as an output of the recurrent layer.

The final output of the recurrent layer of bidirectional RNN contains the past and future information which helps to remember long-term information. For this study, Bidirectional LSTM and bidirectional GRU networks are proposed since both LSTM and GRU networks work to eliminate the long-term dependency problems in simple RNNs.Architectures of both BiLSTM(**Fig.4**) and Bi-GRU(**Fig. *6***) are the same except the recurrent layer is changed that is LSTM and GRU network respectively. The other layers, embedding layer, spatial dropout, dense layer, and output layers are the same for both networks.

Embedding Layer: For the embedding layer, word embedding is used to represent the uniquely extracted words. The word embedding with 300 dimensions is applied for this study. Also, the prepared word2vec is trained by using the same 300 dimensions so that it is supported by this embedding layer. The embedding layer is similar for both BiLSTM and Bi-GRU models.

Recurrent Layer: BiLSTM and BiGRU is used in the recurrent layer for BiLSTM and BiGRU respectively. The bi-directional LSTM is obtained by adding the LSTM layer in forwarding and backward directions to understand the past and future information. The operation is the same for BiGRU unless the GRU gates are used instead of LSTM in the case of bidirectional GRU.

Spatial Dropout layer: SpatialDropout1D is used after embedding layer. Dropout and SpatialDropout1D come to the picture when the neurons in the embedding layer are correlated. Dropout randomly removes elements in the embedding. But Dropout is not enough since the other information is still correlated even if some elements dropped. SpatialDropout1D drops the 1D feature maps from embedding rather than dropping some feature elements. In our case also using SpatialDropout1D in addition to Dropout is very important since there are many redundant and less informative features or words in the dataset. In such a case, the SpatialDropout1D layer drops those 1D features which help to minimize the complexity of the dataset and also has a great role in enhancing the model performance. For example if you see the input layer of BiLSTM and Bi-GRU given below, there are words like malee/እንጅህ and hundi/ሁሉም are less informative relative to the other words in the sentence. So, the spatial dropout will drop such features and so that more informative words only fed in to the recurrent layer. Dropout is also used after the recurrent layer to disable some feature elements before being fed into the dense layer. Finally, the sigmoid layer is applied to classify the data into two classes as “Hate”, and “Free”.

Output layer: After the dropout and dense layers the sigmoid layer is used for the output prediction. The sigmoid function predicts value 0 or 1 and it is used for binary classification. Our system has binary classes and hence, we applied sigmoid for the output layer. The sigmoid function works using the equation:

$\sigma(x)=1/(1+e^{-x})(1)$

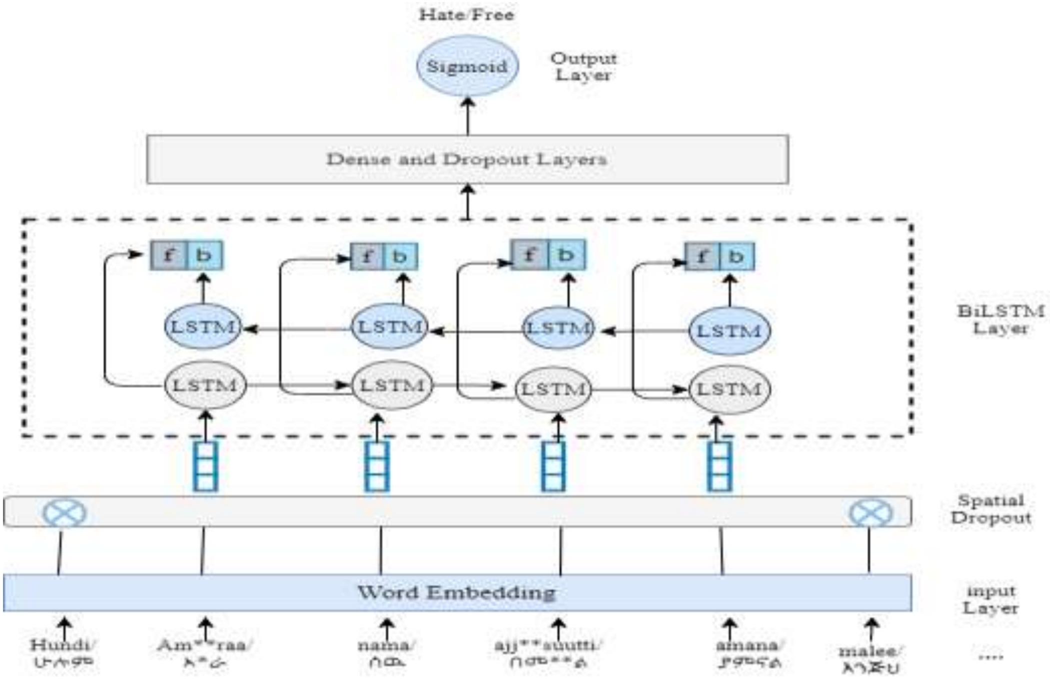


Fig.4.The Proposed BiLSTM Architecture

The recurrent layer of the BiLSTM is combination of LSTM cell. Each LSTM cell contains forget gate (f_t), input gate (i_t), and output gate (o_t). Gates are used to let optionally information through.

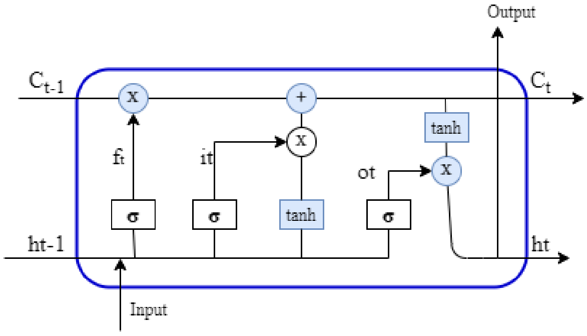


Fig.5. LSTM Cell

Each gate composed out of sigmoid layer and pointwise multiplication. First, the forget gate looks at the ht-1 and the input xt to decide the information to be thrown away from the cell state. Then it gives output 0 meaning throw, or 1 to keep the information. It is calculated as:

f_t=σ(W_f.[h_{t-1}, x_t]+b_f)(2)

Then, the information to be stored in the cell state is decided by the input gate. It identifies the value that will be updated. Then the tan$_h$ creates $\hat{C}_{t.}$ i.e., the new candidate to be added to the cell state. The input gate it and the candidate $\hat{C}_t$ calculated as:

i_t= σ(W_i.[h_{t-1}, x_t]+b_i)(3)

$\hat{C}_t$= σ(W_c.[h_{t-1}, x_t]+b_c) (4)

Then, the old sate ct-1 is updated to ct. by forgetting the information erlier decided to, ft is multiplied by the ct-1 and add multiplication of it and $\hat{C}_t$ to get the new cell state.

C_t=f_t*C_{t-1} + i_t* $\hat{C}_t$(5)

Finally, the last step is deciding what should be the output which is done by the output gate.

O_t= σ(W_o.[h_{t-1}, x_t]+b_o)(6)

The final output layer is obtained from the tanh layer. Which ish_t=ot*tanh(C_t).

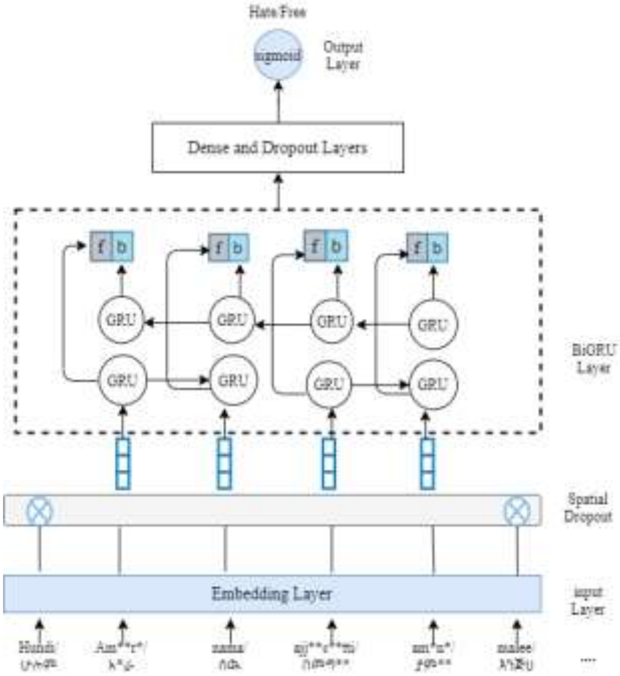


Fig. 6. The proposed Bi-GRU Architecture

The recurrent layer of the Bi-GRU is combination of GRU cell. The GRU cell simplifies the LSTM by merging forget gate and input gate into update gate $z_{t,}$ which controls the forgetting factor and the decision to update the state unit. It contains also the reset gate r_t that controls which parts of the state get used to compute the next target state.

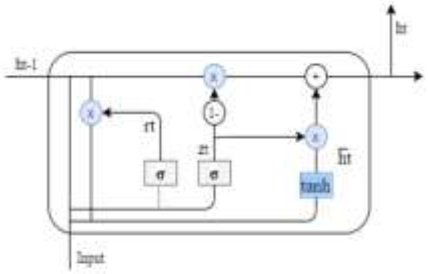


Fig.7. GRU Cell

First, calculate z_t at timestamp t.Then, how much past information to be forgotten is decided by the reset gate r_t. After that, the new content to be stored is calculated and the final output h_t is calculated by the tanh layer.

z_t= σ(W_z.[h_{t-1}, x_t]+b_z)(7)

r_t= σ(W_i.[h_{t-1}, x_t]+b_i)(8)

h_t=tanh(W.[$r_{t*}h_{t-1}$, x_t]+b)(9)

$\overline{h}$t= σ(1-z_t) * h_{t-1} + zt * h_t(10)

V. ATTENTION MECHANISM

The layers of the proposed attention mechanism models are the same as the BiRNNs. But, the attention layer is added after the recurrent layer.[17] presented attention mechanism as a solution for the problem of the encoder-decoder model. In the case ofthe encoder-decoder model, the input is input sequences to one fixed-length vector by the input network, and based on that the output is decoded.However, when we need to focus only on a certain set of inputs that are responsible to provide a particular output encoder-decoder model cannot do that.This is the case why we need an attention mechanism. The attention mechanism is useful to focus on different positions of a single sequence [18]. The attention layer is added in between the encoder layer and the decoder layer to calculate the importance of each input and focus on certain inputs with high weight.

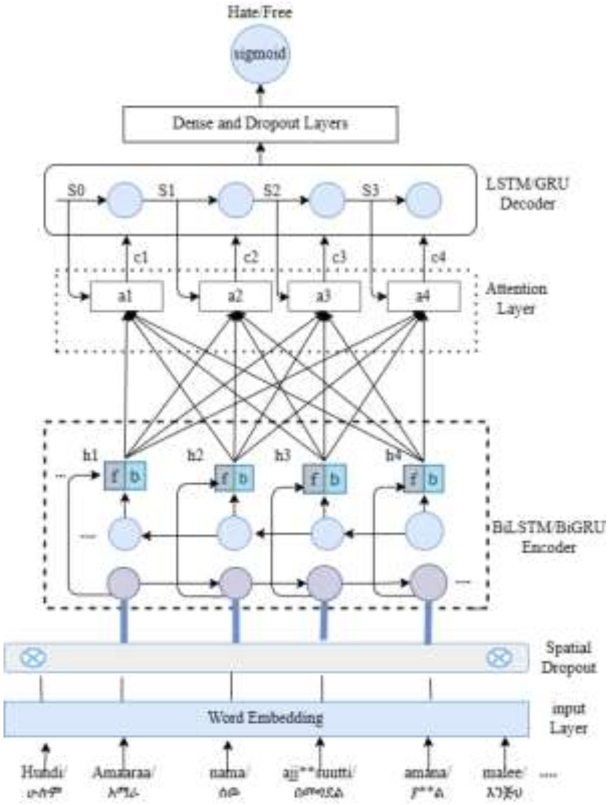


Fig.8. The Proposed Attention Mechanism

The encoder layer is the network accepting input values which is a bidirectional LSTM and GRU in our case. The decoder layer accepts the output of the attention layer and providesan output based on the learned context. In a simple encoder-decoder model the decoder layer decides the final output only based on the output from the encoder layer. The attention mechanism works in such a way that the model can identify what kind of inputs are responsible to give a particular output. The main function of the attention layer is to generate context vectors.The attention layer contains the following three main processes:

- ✓ Alignment
- ✓ Weighting (softmax calculation)
- ✓ Context vectors generation

Alignment:in this step, the output of the decoder layer passed onto perhaps more layers before outputting the final prediction. The alignment operation is putting together the output of the decoder layer with the input of the attention layer hj which are words and output from the recurrent layer. S0, s1, s2, s3.. represents the feedback of the decoder layer and hj represents the h1, h2, h3,…h_{Tx} word features. Based on its importance for providing the output of the decoder the input hj is aligned with the output of the decoder layer. The value of “e” represents the attention annotation and it can be calculated by using the following equation:

$$\mathrm{eij} = \mathrm{a}(\mathrm{Si} - 1, \mathrm{hi})(11)$$

where:

- a represents attention
- s_{i-1}output from the decoder from the previous output time step
- h_i is the output of the decoder layer

Weighting (softmax calculation): Allows threatening scores like probabilities indicating the likelihood of each encoded input time step being relevant to the current decoder output**.** It is known as weight annotation.The link from the output of encoder layers h1, h2, h3, …, $h_{Tx}$to the attention a1, a2, a3,..at is assigned a weight α. For example, for the link from h1 to a1 the αij is α11 and so on. The h1, h2, h3…,h_{Tx} are word features obtained from the recurrent layer. Generally, the weight is represented as αij and it is calculated as:

$$\alpha\mathrm{ij} = \frac{\exp\,(\mathrm{eij})}{\sum_{\mathrm{k=1}}^{\mathrm{Tx}} \exp\,(\mathrm{eik})}(12)$$

Where:T_x is the number of attention inputs or the number of inputs to be focused on by the neural network

Context vectors: Context vectors are the word features we get from the attention layer after calculating their importance**.** This is a weighted sum of the annotations and normalized alignment scoresand itcan be calculated using Equation**Error! Reference source not found.**)

Where:h_i is the output from the encoder layer

VI. RESULTS AND DISCUSSION

The experiment involves developing six models and evaluating the models. We evaluated each model using 5-fold and 10-fold cross-validation to see the performance of models under both 5-fold and 10-fold testing. In addition to the proposed bidirectional recurrent neural networks and attention mechanism, two models LSTM and GRU are implemented and evaluated. The LSTM and GRU models are implemented and tested to compare their performance with the proposed models for the proposed bilingual hate speech detection problem. The word2vec model is applied for the embedding Table ***8***.
layer. The experiment results are given inTable 7 and

Table 7. Model Test Results Using 5-Fold Cross Validation

Model Test Results Using 5-Fold Cross-Validation				
Models	Evaluation Metrics			
	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
LSTM	94	94	94	94
GRU	94	94	94	94
BiLSTM	**94.3**	**94.2**	**94.2**	**94.2**
Bi-GRU	94.2	94	94	94
BiLSTM+Attention	94	94	94	94

Bi-GRU+Attention	94.21	**94.1**	94.1	**94.1**

Table 8: Model Test Results Using 10-Fold Cross Validation

Model Test Results Using 10-Fold Cross-Validation				
Models	Evaluation Metrics			
	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
LSTM	93.8	93.992	93.992	93.992
GRU	94.007	94	94	94
BiLSTM	**94.2**	**94.16**	**94.16**	**94.16**
Bi-GRU	94.1	94.105	94.105	94.105
BiLSTM+Attention	93.82	93.825	93.825	93.825
Bi-GRU+Attention	94.025	94	94	94

The experiment results confirm that using 5-fold cross-validation is slightly better than that of 10-fold for our problem. The test result using 10-fold cv shows that the BiLSTM performs best with an accuracy of 94.2 and an F1-score of 94.16.The second-best performing model is Bi-GRU with an accuracy of 94.1 and an F1-score of 94.105. The performance of the BiLSTM and BiGRU with attention mechanism didn't show any performance improvement. The accuracy and F1-score of BiLSTM are minimized to 93.8 and 93.899 respectively when adding attention mechanism to the model. For the Bi-GRU model when adding the attention layer, the performance is comparable and didn't show a big difference, but still, the performance didn't exceed the BiGRU without attention. Further, we observed that using basic LSTM and GRU models cannot perform better than the proposed bidirectional RNNs and attention mechanism for the proposed system.

The test results using 5-fold cross-validation also show that the performance of BiLSTM exceeds the other models. Except for the GRU and BiLSTM+attention, all the other models performed better. The BiLSTM model accuracy and F1-score are improved to 94.3% and 94.2% respectively. The Bi-GRU+attention model is the second-best performing with an accuracy of 94.21% and an F1-score of 94.1%. Also, the result shows the performance of BiLSTM and BiGRU beats that of LSTM and GRU under 5-fold testing. Generally, the experiment confirms that the proposed bidirectional RNN models scored promising results than basic LSTM and GRU using both 5-fold and 10-fold testing.Further, using 5-fold testing also made us advantageous by improving model performance relative to 10-fold testing.

BILSTM Confusion Matrix

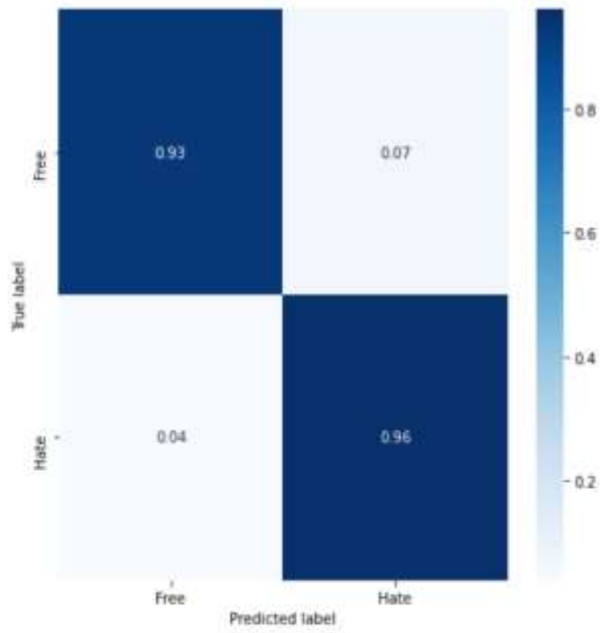


Fig.9. BiLSTM Confusion Matrix

The confusion matrix given in **Fig.9**shows that the BiLSTM model correctly classifies 47,221 instances of the total 50,000 instances in the dataset and only 2,779 instances are misclassified.From the total of 25964 hate instances in the dataset, 96% are classified as true positive and 4% are false negatives. But, for the class “Free” 7% of the total 24036 instances are classified as false negatives and the left 93% of class “Free” instances are correctly classified.

A) Bi-GRU Confusion Matrix

The otherproposed recurrent neural network model we implemented is bidirectional GRU.As shown in **Fig.10**, the model classified 97% of the “Free” instances as true positives and 3% of them as false negatives. For the “Hate” instances, 92% of them are classified as true positives and 8% as false negatives from the total 25964 instances of “Hate” in the dataset.

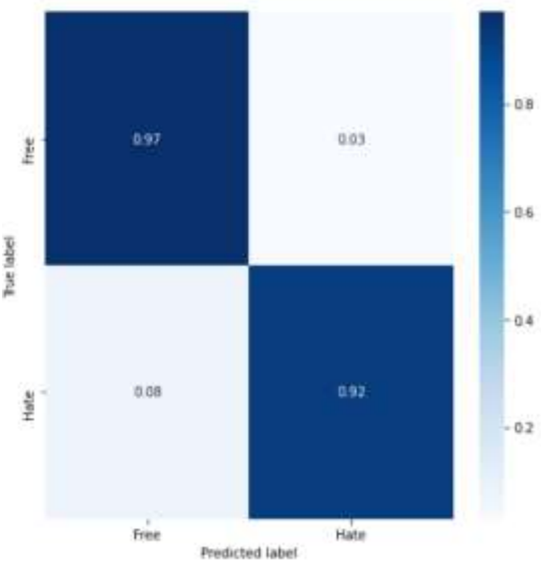


Fig.10. Bi-GRU Confusion Matrix

Generally, the model correctly classified 47,202 instances from the whole dataset.The model is a little bit more confused than the bidirectional long short-term memory as it misclassified 2798 instances while the BiLSTM model misclassified only 2779 instances.

B) Performance of Attention Based Models

After implementing BiLSTM and BiGRU, we applied the attention mechanism by adding an attention layer to both BiLSTM and BiGRU.Using BiLSTM+Attention, 97% of the total 25964 "Hate" instances in the dataset are classified correctly and the left 3% are false negatives as shown in **Fig.11**. But, for "Free" instances, only 91% are true positives, and the rest 9% are false negatives as classified by the BiLSTM+Attention model.

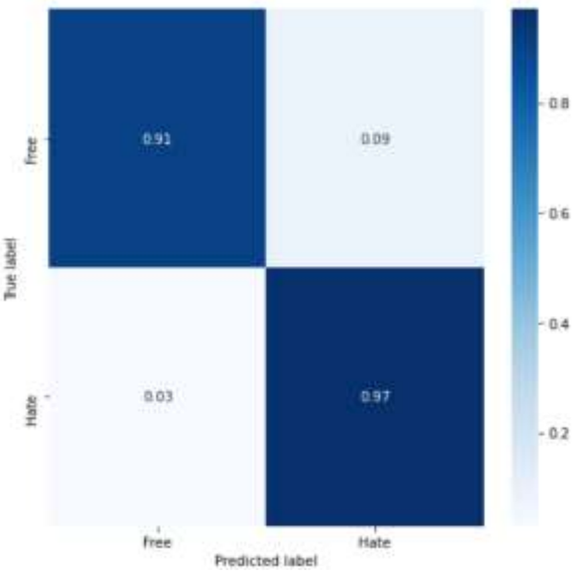


Fig.11.Confusion Matrix for BiLSTM+Attention

The Bi-GRU+Attention model is the second-best performing model when evaluated by 5-fold cross-validation next to the BiLSTM model. The model classified 96% of the total 24,036 "Free" instances in the dataset as true positives and only 4% of them are classified as false negatives. But, for the class "Hate" from the total of 25,964 instances, 93% of them are classified as true positives and the rest are false negatives.

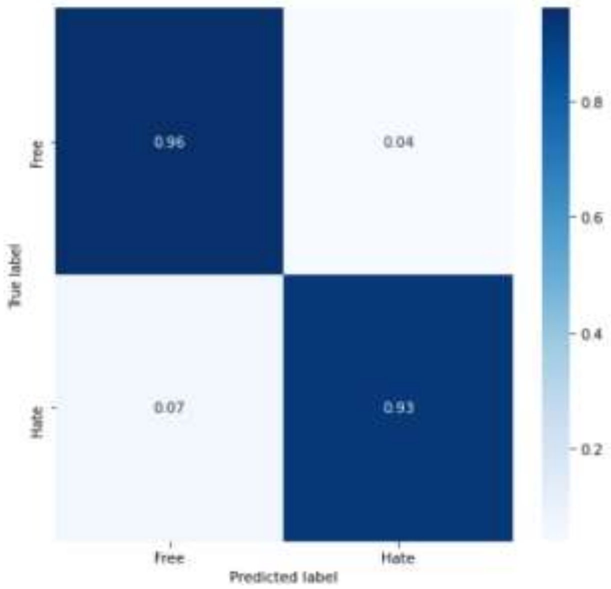


Fig.12. Confusion Matrix for Bi-GRU+Attention

To compare the performance of the proposed models with and without the proposed skip-gram word2vec model, the models are trained also without the word2vec. The overall summary of classification performances of the proposed models using word2vec and without word2vec in terms of accuracy, F1_score, precision, and recall based on 5-fold testing is given inTable 9.

Table 9. Summary of Models Performance Using Word2vec and Without Using Word2vec

Models Performance using Word2vec				
Models	Accuracy	Precision	Recall	F1-score
BiLSTM	**94.3**	**94.2**	**94.2**	**94.2**
Bi-GRU	94.2	94	94	94
BiLSTM +Attention	94	94	94	94
Bi-GRU+Attention	**94.21**	**94.1**	**94.1**	**94.1**
Models Performance Without Using Word2vec				
BiLSTM	94.2	94	94	94
Bi-GRU	94	94	94	94
BiLSTM+Attention	93.89	93.89	93.89	93.89
Bi-GRU+Attention	94	94	94	94

The experiment results presented inTable 9, confirm that applying the proposed word2vec model for feature representation slightly improved the performance. FromTable 9, we observed that BiLSTM with word2vec performance is better than the other models.The BILSTM best performed with an accuracy of 94.3% and the model BiGRU+Attention achievedan accuracy of 94.21%, which is the second-best performing model.

C) Hyperparameter Tuning

The main goal of tuning hyperparameters is to find the optimum hyperparameter that results in low validation error and high model performance.In this study, we performed hyperparameter tuning to find optimal hyperparameter combination for our proposed models. Hyperparameters can be tuned automatically but, it needs huge computational resource. Hence, we used manual hyperparameter tuning. Manual hyperparameter tuning is a trial-and-error process. Since there is no efficient algorithm to select optimum hyperparameter the process requires adjusting several trials to find the parameter combination that provides low validation error.Manually we adjusted hyperparameters and prepared 15 trials for the BiLSTM model. The BiLSTM model is trained and tested on the 15 trials of the parameter combinations. We observed that changing the number of batch sizes and increasing the number of epochs did not show performance improvement. But, selecting the right optimizer and learning rate value shows a significant difference in model performance. For example, using SGD and RMSprop optimizers by adjusting the other parameter the same, the SGD performs very lower than that of RMSprop. Learning rate also has a great role in performance improvement. Selecting an optimal learning rate like 0.001, show good performance rather than using too small a value or large value for learning rate.

Table 10. Hyperparameter tuning trials

Trial	**Hyperparameters**				
	Dropout	Optimizer	Batch Size	Epochs	Learning Rate
1	0.5	RMSprop	256	30	0.002
2	0.5	RMSprop	64	15	0.001
3	0.4	RMSprop	256	10	0.0001
4	0.5	Adamax	128	20	0.0001
5	0.5	Adamax	256	30	0.002
6	0.2	Adamax	128	5	0.1
7	0.5	Adam	256	15	0.01
8	0.4	Adam	32	10	0.1
9	0.5	Adam	256	30	0.002
10	0.5	Adadelta	128	7	0.01
11	0.2	Adam	128	10	0.0001
12	0.4	Adadelta	64	15	0.01

13	0.5	Adadelta	256	30	0.002
14	0.5	SGD	256	25	0.00001
15	0.2	SGD	128	10	0.0001

As shown in (Fig.13), trial 1 and trial 9 achieved performance first obtained by the BiLSTM model. But, no trial results exceed the performance first achieved by the BiLSTM model.

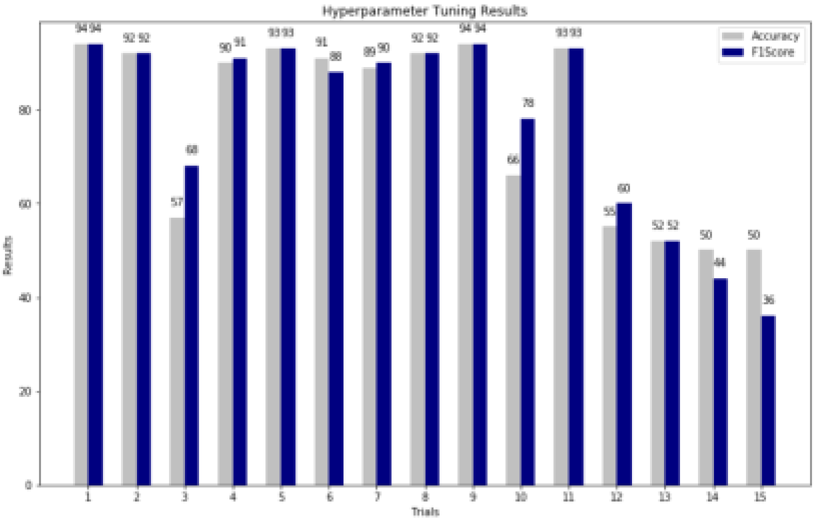


Fig.13.Hyper parameter Tuning Results

In addition to the BiLSTM model, the BiGRU+Attention is also implemented by using the parameter combination of trial 1 and trial 9 for model comparison. The BiGRU+Attention model is the second-best performing model.

Table 11. Comparison of BiLSTM with BiGRU+Attention

Models	**Trial 1**		**Trial 9**	
	Accuracy(%)	F1-score(%)	Accuracy(%)	F1-score(%)
BiLSTM	**94**	93.846	**94.2**	94.2
BiGRU+Attention	93.56	93.56	94	94

From the comparison of BiLSTM and attention-based BiGRU models, we observe that the performance of the BiLSTM model stayed unbeaten by BiGRU+Attention. Finally, the experiment confirms that the BiLSTM model performs best for the proposed bilingual social media hate speech detection on the prepared bilingual dataset.The better performance achieved by BiLSTM is obtained by training the models using the following hyperparameter combinations:

Table 12. Hyperparameter combinations used for model training

Hyperparameter names	Values

Dropout	0.5
Optimizer	Adam
Batch Size	128
Epochs	30
Learning Rate	0.001

D) Model Testing Using New Input Data

To evaluate the model prediction,new posts and comments are collected. The data for evaluation is collected from five peoples where each of them are instructed to prepare 10 instances; five Afaan Oromo and five Amharic, from Facebook posts and comments. As a result, we got 50 new instances.We observed that only five instances are misclassified by the model after checking the model prediction for all 50 instances.Observing this, we tried error analysis to identify the reason for model misclassification.There are some reasons for misclassification that we identified. The first challenging problem we observed is that peoples use the name of ethnic group, name of religion, and other names of the protected category to offend the targeted group when they post hate speech. Therefore, most of the hate speech texts in the training data contain the name of the protected identity of the targeted group.

Hence, the model sometimes classifies such texts as hate speech. The other one is misclassification due to abusive words in the text. Peoples use abusive words mostly to offend the targeted group. The hate speech dataset contains a lot of abusive words which are labeled as hate speech. Due to that the model mostly classifies text containing abusive words as hate speech even if it is written in another context. The dataset is also limited to social media data. Therefore, lack of a huge dataset other than social media data is another problem making the model unable to correctly classify some instances.

E) Discussions

The main goal of this study is to develop a deep learning model for bilingual hate speech detection for Afaan Oromo and Amharic texts. To the best of our knowledge, this study is the first to propose bilingual social media hate speech detection for Afaan Oromo and Amharic languages using deep learning approaches. To implement the models, we used a bilingual dataset prepared from newly collected Afaan Oromo texts from the Facebook platform and the existing Amharic hate speech dataset.Before the model implementation, we applied text preprocessing including cleaning and normalization. We normalized Amharic characters those having the same meaning but spelled differently. We used word embeddings and we trained word2vec for the feature representation to develop the models.

Six models are implemented and evaluated using the prepared bilingual dataset.Looking for the advantages of using bidirectional RNN models to store past and future information, we implemented BiLSTM and BiGRU models. In addition to that, attention-based BiLSTM and attention-based BiGRU are implemented expecting the advantages of using an attention mechanism that helps to calculate the significance of each input as stated in [19]. The LSTM and GRU models are also implemented to compare their performance with the proposed models. First, the models are tested using 5-fold and 10-fold cross-validation. Best model performance is obtained using the BiLSTM model using 5-fold evaluation. The experiment shows that using 5-fold is better than using 10-fold for our bilingual dataset.The experiment results using 5-fold testing shows that the BiLSTM model with word2vec performed better on our bilingual dataset (Table 9). The attention-based BiGRU model is the second-best performing model. For the BiLSTM adding the attention layer, unable to outperform the BiLSTM baseline while the BiGRU+attention performed slightly better than the BiGRU. Based on the evaluation results, the BiLSTM performed better than all the other models.

For performance improvement and to mitigate the problem of overfitting we used SpatialDropout1D and a Dropout rate of 0.5. Also, EarlyStopping is applied during model training to enabling the model to stop learning at optimum epochs.Also, we prepared a trained word2vec model based on the skipgram model. The models are implemented using the trained word2vec and without using the trained word2vec, the result shows that the developed word2vec model enhanced the model performance slightly (Table 9). Further, hyperparameter tuning is performed to come up with the optimum hyperparameter combination providing high accuracy and low validation error. To do that, the selected BiLSTM is tested on 15 trials of parameter combination. Since hyperparameter tuning requires more GPU devices and is time taking process only 15 trials are manually adjusted. By testing the model performance using the 15 parameter combination trials, no trial result exceeds the performance first obtained by the BiLSTM model. But the best performance is obtained using trials 1 and 9 which is accuracy and f1_score of 94%, and that is the same asthe performance first achieved by the BiLSTM model.

To compare the model performance, the BiGRU+attention model is selected, which is the second-best performing model and compared with the BiLSTM model. The attention-based BiGRU is trained using trial 1 and trial 9 parameter combinations. The result shows that the performance of theBiLSTM model stays better for the prepared bilingual dataset.

Finally, the accuracy of 94.3% and an f1 score of 94.2% achieved by the BiLSTM model stayed the best performance for the proposed bilingual social media hate speech detection for Afaan Oromo and Amharic languages. The prototype is developed using the BiLSTM model.

We developed GUI for the prototype and deployed the model on a webserver using FLASK API. Then, the error analyses process is performed based on newly collected 50 instances to testmodel predictions using new input data and identify the reason for misclassification. Most misclassification problem is related to the effect of abusive words. Since the hate speech dataset contains a lot of abusive words the model mostly understands any text with abusive words as hate speech. Also, the other big challenge is there are names of the protected identity that come with hate speech in the training data. Hence, the model understands and classifies such words as hate speech. Classifying hate speech correctly requires huge training data with correct labels.

VII CONCLUSION AND FUTURE WORKS

This study proposed bilingual social media texts hate speech detection for Afaan Oromo and Amharic languages using deep learning approaches.To accomplish this work first the bilingual dataset is prepared using Afaan Oromo and Amharic texts. Different preprocessing techniques are applied to the dataset before representing it as a feature vector.Word embedding is used for feature representation to develop the proposed models.We prepared also a trained word2vec using the Amharic and Afaan Oromo texts. The word2vec is trained using the skip-gram model. For the experiment, six models are developed using the bilingual dataset. The LSTM and GRU algorithms are implemented in addition to the proposed bidirectional RNNs and attention mechanism for model comparison. Also, the models are compared by using the trained word2vec and by training models without using our trained word2vec. The BiLSTM model with word2vec outperformed all models by evaluating models using 5-fold cross-validation. The BiLSTM model achievedaccuracy and f1-score of 94.3% and 94.2% respectively. Despite its advantages of focusing on important keywords, adding the attention mechanism to the BiLSTM model is unable to outperform the BiLSTM baseline. But, the Bi-GRU model with an attention mechanism shows slightly better performance than the BiGRU baseline. The experiment shows that the Bi-GRU with attention mechanism performed best next to the BiLSTM model while the performance of the LSTM and GRU was unable to beat the proposed BiRNNs and attention mechanism.

Hyperparameter tuning is performed by adjusting 15 trials for hyperparameter combinations. The tuning process is done for the BiLSTM model which outperformed the others. Even if we did not get significant improvement, two trials able to achieve comparable performance to the performance first obtained by the BiLSTM. Finally, this study concludes that BiLSTM is the best performing model for the proposed bilingual hate speech detection of Afaan Oromo and Amharic language texts using our dataset. Also, the study proved that using pre-trained word2vec models is useful for performance improvement.

Although this study implemented and conducted an experiment on recurrent neural networks and created a baseline attention mechanism for bilingual hate speech detection, future research can include custom large pre-trained word embedding using social media data for further performance improvement.Since deep learning models like CapsNet also achieving good performance in NLP tasks, ensemble deep learning models can be implemented for the problem by integrating the BiLSTM model with those models.

Further, the BiLSTM model can be tested on non-textual data such as audio and video by future researchers for the problem of Afaan Oromo and Amharic language hate speech detection.

Data Availability

The Afaan Oromo data was collected from different public Facebook pages and merged with the Amharic dataset to build the bilingual dataset.The Amharic language dataset used for this study is downloaded from DOI: 10.17632/ymtmxx385m.1.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgement

This paper was done as a thesis by sponsorship of Adama Science and Technology University (ASTU), Ethiopia.The Thesis is available on etd.astu.edu.et. We want to acknowledge ASTU for helping us to conduct this research.

REFERENCES

[1] H. A. Nayel and H. L. Shashirekha, “DEEP at HASOC2019 : A machine learning framework for hate speech and offensive language detection,” CEUR Workshop Proc., vol. 2517, no. December 2019, pp. 336–343, 2019.

[2] S. S. Aluru, B. Mathew, P. Saha, and A. Mukherjee, “Deep Learning Models for Multilingual Hate Speech Detection,” pp. 1–16, 2020, [Online]. Available: http://arxiv.org/abs/2004.06465.

[3] Z. Mossie and J.-H. Wang, “Social Network Hate Speech Detection for Amharic Language,” pp. 41–55, 2018, doi: 10.5121/csit.2018.80604.

[4] Z. Zhang, “Hate Speech Detection : A Solved Problem ? The Challenging Case of Long Tail on Twitter,” vol. 1, no. 0, pp. 1–5, 1900.

[5] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, “Deep Learning for Hate Speech Detection in Tweets ∗,” no. August 2020, pp. 7–9, 2017, doi: 10.475/123.

[6] P. Fortuna and I. Tec, “A Survey on Automatic Detection of Hate Speech in Text,” vol. 51, no. 4, 2020.

[7] Y. Kenenisa and T. Melak, “Hate Speech Detection for Amharic Language on Social Media Using Machine Learning Techniques,” ASTU, 2019.

[8] E. Baweke, “AMHARIC TEXT HATE SPEECH DETECTION IN SOCIAL MEDIA USING DEEP LEARNING APPROACH,” BAHIR DAR UNIVERSITY, 2020.

[9] A. Gaydhani, V. Doma, S. Kendre, and L. Bhagwat, “Detecting Hate Speech and Offensive Language on Twitter using Machine Learning : An N-gram and TFIDF based Approach,” 2018.

[10] T. L. Sutejo and D. P. Lestari, “Indonesia Hate Speech Detection Using Deep Learning,” Proc. 2018 Int. Conf. Asian Lang. Process. IALP 2018, pp. 39–43, 2019, doi: 10.1109/IALP.2018.8629154.

[11] S. G. Tesfaye and K. K. Tune, “Automated Amharic Hate speech Posts and Comments Detection Model using Recurrent Neural Network,” 2020.

[12] F. Del Vigna, A. Cimino, F. Dell’Orletta, M. Petrocchi, and M. Tesconi, “Hate me, hate me not: Hate speech detection on Facebook,” CEUR Workshop Proc., vol. 1816, no. January, pp. 86–95, 2017.

[13] I. Aljarah et al., “Intelligent detection of hate speech in Arabic social network: A machine learning approach,” J. Inf. Sci., 2020, doi: 10.1177/0165551520917651.

[14] H. Faris, I. Aljarah, M. Habib, and P. A. Castillo, “Hate speech detection using word embedding and deep learning in the Arabic language context,” ICPRAM 2020 - Proc. 9th Int. Conf. Pattern Recognit. Appl. Methods, no. Icpram, pp. 453–460, 2020, doi: 10.5220/0008954004530460.

[15] G. O. Ganfure, “Comparative analysis of deep learning based Afaan Oromo hate speech detection,” J. Big Data, vol. 9, no. 1, 2022, doi: 10.1186/s40537-022-00628-w.

[16] A. Rushdy, “Aadhil Rushdy Word Embeddings for Sentence Classification,” pp. 1–6, 2021.

[17] D. Bahdanau, K. H. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–15, 2015.

[18] A. Vaswani et al., “Attention is all you need,” Advances in Neural Information Processing Systems, 2017. .

[19] X. Sun and W. Lu, “Understanding Attention for Text Classification,” no. 1999, pp. 3418–3428, 2020, doi: 10.18653/v1/2020.acl-main.312.