Received: 8 August 2023 | Revised: 19 October 2023 | Accepted: 23 November 2023 | Published online: 23 November 2023

Journal of Data Science and Intelligent Systems
2025, Vol. 3(1) 67–77
DOI: 10.47852/bonviewJDSIS32021512

RESEARCH ARTICLE

BON VIEW PUBLISHING

Multichannel 2D-CNN Attention-Based BiLSTM Method for Low-Resource Ewe Sentiment Analysis

Victor Kwaku Agbesi[1,*], Wenyu Chen[1,*], Chiagoziem C. Ukwuoma[2], Noble A. Kuadey[1], Collinson Colin M. Agbesi[3], Chukwuebuka J. Ejiyi[1,4], Emmanuel S. A. Gyarteng[1], Gladys W. Muoka[1,4] and Anthony M. Kuadey[5]

[1]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, China*

[2]*College of Nuclear Technology and Automation Engineering, Chengdu University of Technology, China*

[3]*Computer Science and Engineering, Koforidua Technical University, Ghana*

[4]*School of Information and Software Engineering, University of Electronic Science and Technology of China, China*

[5]*Department of Mathematics/ICT, St. Francis College of Education, Ghana*

Abstract: The unavailability of an annotated dataset for a low-resource Ewe language makes it difficult to develop an automated system to appropriately evaluate public opinion on events, news, policies, and regulations. In this study, we collected and preprocessed a low-resourced document-level Ewe sentiment dataset based on social media comments. We used three features learned by word embeddings (Global vectors, word-to-vector, and FastText) rather than hand-crafted features. We further proposed a novel method termed MC2D-CNN+BiLSTM-Attn to detect the exact sentiment feature from the Ewe dataset. Extensive experiments indicate that the proposed method efficiently classifies various sentiments and is superior to benchmark deep learning methods. Results show that in detecting the precise sentiments from raw Ewe textual context, the BiLSTM incorporating Glove outperforms Word2Vec and FastText embedding with an accuracy of 0.727. Furthermore, Attn +BiLSTM and multichannel convolutional neural network methods incorporating the Word2Vec embedding layer perform better than Glove and FastText embedding with an accuracy of 0.848 and 0.896. In contrast, our proposed method with the same Word2Vec embedding recorded 0.949.

Keywords: document level-sentiment analysis, EweSentiment, Multichannel CNN, natural language processing

1. Introduction

Sentiment analysis is a powerful tool for learning individual reactions, thoughts, and feelings about an event, a specific topic, or a product [1]. Natural language processing (NLP) is characterized as the procedure of interpreting subjective texts with emotional undertones, such as insightful comments about individuals, items, policies and regulations, and events, uploaded by a person on the web and then processed to extract exact sense out of them. With the rapid advancement of technology, social networking has been well-embraced in the world of communication. The interconnectivity of the world paves the way for many people to be connected throughout the world on the backbone of internet-based media technologies. As a result, more than 85% of people worldwide use social media platforms (including Twitter, WhatsApp, Facebook, LinkedIn, Instagram, and YouTube) as a primary tool to express their views, likes, dislikes, or opinions on almost anything in the form of texts, audio-visuals, and so on [2]. Sentiment is the most important factor in human textual communication on social media. People utilize text messages these days to share thoughts, rate their preferences, offer input for various services, evaluate, and, in some cases, propose products. This created a plenitude of handy information on the web and developed smart mechanisms to aid institutions in making more enlightened decisions [3].

Several systems have been effectively deployed to aid users in diverse ways. These systems typically categorize sentiments using a vast range of methods, such as support vector machines [4], decision trees [5], and Bayes classifiers [6]. However, these methods are associated with several known flaws that cannot be disregarded despite their simplicity, high performance on tiny datasets, and cheap system needs. They mostly depend on manual extraction, resulting in inadequate and imprecise emotional expression, limited capacity to describe complicated functions, and difficulty handling complex categorization problems. Before the effective implementation of deep learning (DL) architecture, a significant amount of effort had been devoted to identifying methods for enhancing sentiment analysis capability. Recurrent neural networks (RNNs) and convolutional

*Corresponding authors: Victor Kwaku Agbesi, School of Computer Science and Engineering, University of Electronic Science and Technology of China, China. Email: vkagbesi@std.uestc.edu.cn and Wenyu Chen, School of Computer Science and Engineering, University of Electronic Science and Technology of China, China. Email: cwy@uestc.edu.cn

neural networks (CNNs) were frequently used. Document-level sentiment analysis (DLSA) tools based on CNN architectures extract features using a weighted network, reducing the requirement for hand-crafted features. Significantly, the network is simple but operates simultaneously. In addition, attention frameworks [7] and dilated CNN [8] with embeddings enhance the performance of various sentiment analysis studies. However, orthodox CNNs alone need help to learn from sequential proportion, hence assembling more layers to capture sentiment contexts adequately, while conventional RNNs cannot record long-term dependencies. Given this challenge, long short-term memory (LSTM) and gated recurrent unit [9, 10] were designed to solve the problem. Still, these methods can only solve the long-dependence puzzle for long sequences in DLSA correctly. Further, multilingual studies incorporate hybrid DL [11] and commonsense models such as BabelSenticNet [12] for sentiment resource development. Still, these models often ignore the essential part of most low-resource languages (LRLs) original features.

With the introduction of social media, however, people are increasingly eager to express themselves in their native languages via written text, audio, video, etc. As a result, it is necessary to analyze sentiments in other languages to prevent apathy, loss of essential details that may be offered, promote linguistic development, and ensure language knowledge expansion. Until now, most of the available research on sentiment analysis has focused solely on accessible textual data in English and other high-resource languages, including Chinese, Spanish, and German. However, sentiment analysis and topic modeling (TM) are new ideas for the low-resourced Ewe language that have yet to be researched. The Ewe language faces numerous challenges, including (1) a lack of publicly available resources such as datasets, dictionaries or lexicons, and NLP tools that could be used to solve any associated NLP problem, making the language vulnerable. (2) To determine the Ewe textual sentiment expression, the implied expression by Ewe tutors and native speakers is the best way. (3) Comparable Ewe sentiments are hazy, potentially resulting in collocation errors. (4) Most comments or posts in Ewe contain numerous objectives, and each objective has its sentiment, making it difficult and erroneous to classify the exact sentiment of the entire statement.

Motivated by the methods above and the challenges, this study creates a new Ewe-based sentiment dataset from comments gathered from individuals on various social media platforms. We propose a multichannel two-dimensional CNN fused with an attention-based-bidirectional long-short-term memory (MC2D-CNN +BiLSTM-Attn) method for a DLSA. The framework captures high-level Ewe sentiment features from the newly created dataset. Our multichannel structure efficiently obtains unique, diverse, high-level Ewe context information, which is conducive to learning sentiment features from distinctive alphabetical letters, including ɖ, ʋ, ƒ, í, ŋ, ɔ, ɖ, ɔ̃, ã, ɛ, ʋ, ê, etc. to improve text representation of our method. Below are the main contributions of this study.

1) We collect and preprocess the Ewe-based sentiment dataset, "EweSentiment," which contains individual comments on various social media platforms (Twitter, WhatsApp, Facebook, WeChat, LinkedIn, and Instagram). The EweSentiment comprises 4315 document-level texts on five predefined sentiment topics, including angry, sad, happy, surprised, and annoyed.
2) Based on the newly created EweSentiment dataset, we developed three different word embeddings, such as Glove, FastText, and Word2Vec, for exploiting sentiment representation in the Ewe language.
3) Then, we proposed a novel MC2D-CNN+BiLSTM-Attn to detect the exact sentiment feature from the Ewe document. The multichannel 2D-CNN is composed of various kernel shapes (i.e., 1×1, 3×3, and 5×5), which is conducive to learning high sentiment features of different scales (ɖ, ƒ, í, ŋ, ɔ, ɖ, ã, ɔ̃, ɛ, ʋ, ê) to improve the text representation of the method.
4) We analyze the effects of different learning rates (LRs) on the optimization procedure and present an in-depth error analysis of the proposed method. Extensive simulations indicate the robustness and stability of our proposed MC2D-CNN+BiLSTM-Attn method in learning the Ewe sentiment features. Finally, we agree that sentiment analysis in the Ewe language has a wide jurisdiction of subsequent potential with benchmark methods.

The study is structured as follows. Section 2 briefly reviews literature based on sentiment analysis in LRLs. Details of our dataset construction and formation are presented in Section 3. In contrast, in Section 4, we explain the three embeddings and give the details of our proposed method. Section 5 evaluates and visualizes results for all three benchmark methods, comparing them to the proposed method. Research discussion and implications are presented in Section 6, and finally, conclusion and recommendation of future research direction are given in Section 7.

2. Related Works

Different studies have tried many times to create structured lexicons and datasets, proposed methods with outstanding frameworks, and other linguistic resources to help analyze sentiments in LRLs. However, the complexity of analyzing emotions in most LRLs makes it more difficult for researchers to attempt the study [13].

The authors created a 15K Roman Urdu dataset to solve this challenge using statements and comments from various social media platforms [14]. A lexicon was created for Roman Urdu that assigns an emotional value to terms, similar to the senti-strength framework. There were 91K political tweets, and 21K non-political tweets gathered. In addition, 7186 tweets are classified as Roman Urdu in the linguistic categorization. Mehmood et al. [15] use linguistic features of Roman Urdu texts to transform lexical variation texts into canonical structures.

Works on Bangla sentiment have recently been studied. Sarkar [16] presented a method for recognizing sentiments from Twitter data using deep-CNNs. Different approaches acknowledged multitype abusive Bangla text [17]. Moreover, improved stemming criteria for the Bangla language have been developed, resulting in improved performance. Irtiza Trinto and Eunus Ali [18] obtained data from YouTube comments in Bangla, English, and Romanized Bangla languages and described them based on three sentiments. But Bangla sentences were classified into six different emotions based on the content. However, they have skimmed over the longer remarks. Mahmud et al. [19] used Bangla texts to categorize three sentiments using Glove word embedding, Adam optimizer, CNN's DL classifier, and Glove-BiLSTM. Glove-CNN was introduced to ensure better performance from the banal text data. Tasnim et al. [20] collected and detected depressive posts from Bangla social media texts. Sarkar [21] further compares the performances of DL-based methods, including LSTM, CNN, and BiLSTM, in classifying Bangla texts. A Bangla text dataset was generated based on individual Facebook replies on various sentiment classes [22].

They utilized the newly created dataset to classify predefined emotions using the Nave Bayes classifier. An attention-based CNN is employed for the first time to analyze various sentiments in Bangla with an accuracy of 0.7206 [23]. With advanced layers in the network, the authors achieved a 0.9401 accuracy for LSTM. DL methods have been exploited with two separate Bangla datasets to create potential evaluations between all models [24]. Islam et al. [25] presented a fully tagged 2nd-class and 3rd-class corpora in Bengali content. Specifically, they show how a multilingual BERT model with suitable extensions may be trained across unique datasets using a transfer learning procedure to enhance SOTA performance in sentiment classification tasks. Their model outperforms the current SOTA method with an accuracy of 0.6803 for 2nd-class sentiment categorization. Smetanin and Komarov [26] uncovered a sentiment dataset for the Russian language. They improved multilingual RuBERT, BERT, and two multilingual variants. They found that the Russian text did better than rule-based and basic SOTA methods in classification performance. Similarly, the Ewe dataset was finetuned with BERT for better Ewe-based classification accuracy [27].

Sentiments have been analyzed in other LRLs, including Arabic, Hindi, Khmer, and Thai. ArWordVec [28], a customized Arabic-based word embedding model developed for tweet categorization, is the broadest research in the Arabic language. Rizkallah et al. [29] addressed the drawbacks associated with ArWordVec. In particular, ArWordVec needs to handle Arabic texts with polarities from unrelated word pairs. To this end, they developed ArSphere to solve the Arabic polarity problem. Instead of embedding the vectors onto the entire area, their work is on a unit sphere. The sphere embedding is appropriate because polarity may be dealt with by embedding tensors at the sphere's opposing poles. The suggested strategy offers some benefits. According to them, it fully utilizes rich and complex text tools created by linguists, such as dictionaries from the past. In contrast, the conventional method of building word embedding makes use of the idea of word co-occurrence. Another benefit is that it does a good job of separating synonyms, antonyms, and unrelated word pairings. Alzanin et al. [30] created 35,600 Arabic-based Twitter comments and preprocessed them. They present a method for categorizing written tweets in Arabic into five distinct groups based on their linguistic properties and content. They also investigate two written expressions: word embedding using Word2Vec and stemmed text with term frequency-inverse document frequency using three different methods as their classifiers. Little attention has been given to the Hindi language. The Hindi comments were gathered based on negation and discourse connection [31]. The novel dataset of annotated Hindi reviews was constructed and categorized using the polarity-based method, which recorded a 0.8021 accuracy. A DL framework is proposed to detect exact abusive words in Thai. This method produces an F1-score of 0.8632. Similarly, Al-Amin et al. [32] analyze sentiments in Bengali. They use Bengali comments for their analysis using Word2Vec and a generic sentiment detection procedure. They argue that using the Word2Vec approach provides an essential insight into the Bengali language. Their strategy provided an accuracy of 0.755. On the other hand, the Dengue dataset was utilized to classify Filipino-based sentiment analysis, where authors used DL methods to learn semantic features from the derived texts [33].

Following the above-reviewed studies, it is evident that LRLs, especially African languages, have not been adequately investigated with SOTA methods. As a result, it is critical to study and analyze these techniques with the low-resourced Ewe language. As far as we know, a study has yet to be done with the Ewe text to analyze the sentiments of people.

3. EweSentiment Dataset Construction

This section details all the procedures involved in developing the EweSentiment dataset. This includes a brief dataset description and the data formation and preprocessing phases.

3.1. Dataset description

The Ewe language (Eʋegbe) is part of a group of related languages known as *Gbe*-language in Africa. It is spoken by over 20 million people in Togo, Liberia, Ghana, Benin, Nigeria, and Niger-Congo. The Ewe alphabet contains 30 letters. Social media platforms have many free Ewe-based texts, audio, and videos. However, due to the lack of machine-readable datasets and sentiment lexicons, Ewe language analysis remains challenging. There are further difficulties with the Ewe language, such as irregular syntax and consonant clusters. Furthermore, few resources are available in Ewe; as a result, the Ewe language has received less attention.

In this study, we collected and preprocessed a document-level Ewe sentiment dataset named "EweSentiment" by translating generic comments gathered on various social media platforms (Instagram, WeChat, Facebook, WhatsApp, Linked-In, and Twitter) on five sentiment topics into the Ewe language using the Google Translate Application. This is because, for the low-resource Ewe language, no publicly available dataset could be used for this study; hence, an Ewe-based dataset is needed. In total, 4315 comments from about 178,980 texts were collected based on predefined 5 basic sentiment topics, with each sentiment class of 863 comments from individuals. The sentiment topics are angry, annoyed, happy, surprised, and sad [34]. Examples of the Ewe texts in the newly created EweSentiment dataset with their corresponding English meanings are provided in Table 1. A sample document-level text contains about 125 words.

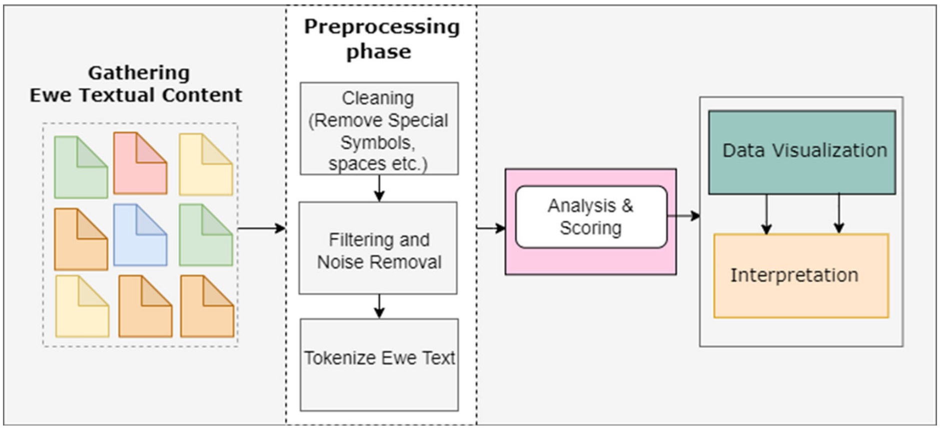
Table 1
Details of the EweSentiment dataset

Sentiment	Ewe text	English meaning	Size
Angry	Edze ƒã be menye ɖe wòkpe wo o	It's clear he had not invited them	863
Annoyed	Mègado dziku gbeɖe ne èsee o ku ɖe alesi wòwɔ wowɔ atike nɛ to eƒe dɔwɔhati	Never get upset when you heard about how he was treated by his colleague	863
Happy	Dzi dzɔm ŋutɔ esi miaƒe ŋutsuwo ƒe dukɔa ƒe bɔlƒoha dze	I jubilated when our men's national team qualified	863
Surprise	Ewɔa nuku nam ŋutɔ	She astonishes me	863
Sad	Dunyahelawo mekpɔnɛ dzea sii kura be yewonye hiãtɔwo o	Politicians don't recognize themselvesas needy at all	863

3.2. Dataset formation and preprocessing

In a typical sentiment analysis procedure, data collection, preprocessing, analysis and scoring, and data visualization and

Figure 1
The general framework of the sentiment analysis procedure



interpretation are the four main phases of performing a sentiment analysis task (Figure 1).

The sentiment texts were collected from individuals on various social media platforms. For the preprocessing phase, we converted all upper-cased letters to lower-cased letters, cleaned (remove noises: hashtags, empty spaces, full stops, punctuation from each Ewe text, except for special alphabetical letters including ɖ, ʋ, ƒ, í, ŋ, ɔ, ɖ, ɔ̃, ã, ɛ, *υ*, ê, etc.), and filtered the EweSentiment dataset. Finally, tokenization techniques were initiated based on each word embedding to represent the Ewe textual content as vectors, as machines only understand numbers (example: 0s and 1s) and not strings of words (e.g., mɔɖaŋudɔwɔla, ŋutinuwo, gadzraɖoƒe) [35]. After this procedure, we recorded a total of 419,867 Ewe-based tokens. As indicated in Table 1, we use the document-level analysis to process the preprocessed dataset because we realize that, based on the Ewe phonics and sentence structure, sentimental expressions are mostly at the end of each text document.

The preprocessed EweSentiment dataset consists of comments ranging from a single word to an average of 125 words. We realize that Irtiza Trinto and Eunus Ali [18] skipped most comments to record a maximum length of 30 words. However, emotions are expressed later in the text in most social media posts, though a single word can detect emotion [36]. For this reason, we encourage all respondents to express their sentiments within a minimum text length of 125 words. The results were visualized, analyzed, and interpreted in the last phase for in-depth understanding.

4. Research Methodology

To efficiently solve the sentiment analysis problem associated with the Ewe language, we modified three SOTA models and proposed a multichannel two-dimensional CNN plus attention-based-bidirectional long-short-term memory (MC2D-CNN +BiLSTM-Attn) method to efficiently learn sentiment features from the newly developed EweSentiment dataset.

As illustrated in Figure 2, the method comprises three main parts: the preprocessing, word embedding, and the proposed ML method. In this study, for the first time, we developed three different word embedding input vectors based on the Ewe text and compared their superiority with all SOTA methods. Furthermore, the proposed method is segmented into two main parts, namely the multichannel two-dimensional CNN (MC 2D-CNN) with diverse convolution kernel sizes and an attention-based BiLSTM mechanism. The core idea for exploring a multichannel 2D-CNN is to adequately capture semantic information and sentiment features that might be made up of some unique alphabetical letters in the Ewe language (such as ɖ, ʋ, ƒ, í, ŋ, ɔ, ɖ, ɔ̃, ã, ɛ, *υ*, and ê). The attention-based BiLSTM mechanism is deployed to boost feature vector dimensions and augment semantic features and information.

Finally, the novel MC2D-CNN+BiLSTM-Attn is used to learn the Ewe sentiment features and improve the extraction capabilities of the Ewe text compared to other benchmark methods. Detailed procedures involved in this study are explained below.

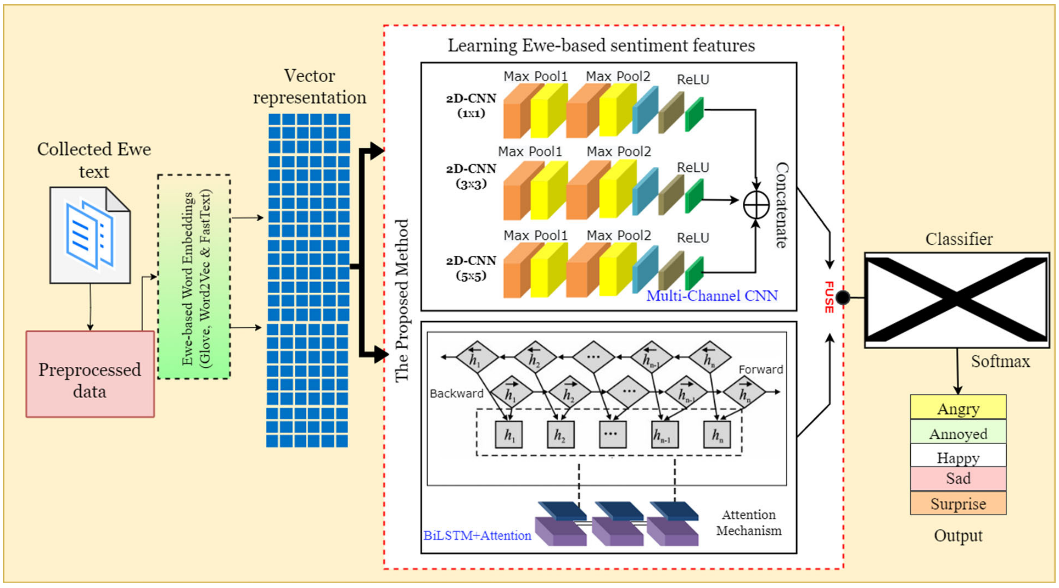
4.1. Word embeddings

Embedding words is one of the most common use practices for feature learning and language modeling in NLP. It is essential for the dense vector representation of words and documents. Several word embeddings have been explored for classification tasks, especially in sentiment analysis and text classification. In this study, we formulate different embedding layers such as Glove [37], FastText [38], and Word2Vec [39] to extract traits from the Ewe text.

4.2. The proposed method

In this study, we proposed an Ewe-based sentiment feature learning method based on a multichannel 2D-CNN and BiLSTM-attention (MC2D-CNN+BiLSTM-Attn) for sentiment analysis. We extract the sentiment features from our Ewe document by integrating these two models. The main reason for using a multichannel 2D-CNN is to adequately capture semantic information and detailed sentiment features formed from some "unique" alphabetical letters, including ɖ, ʋ, ƒ, í, ŋ, ɔ, ɖ, ɔ̃, ã, ɛ, *υ*, ê, etc. in the Ewe text, which is not part of the English alphabets. Also, CNNs are automatic feature learning models particularly effective in capturing spatial hierarchies and local patterns in data, especially in downstream tasks. The convolutional layers automatically learn features from local receptive fields, representing important texture patterns. Additionally, by sharing weights through convolutional layers, CNNs can model local patterns with far fewer parameters, which makes them

Figure 2
The proposed MC2D-CNN+BiLSTM-Attn method



computationally efficient and less prone to overfitting complex and unique texts. On the other hand, the BiLSTM is capable of capturing contextual information, handling variable-length sequences efficiently, and is effective for sequential data like that of Ewe text. BiLSTMs capture the nuanced relationships between words and their surrounding context, which is significant for sentiment classification. Our 2D-CNN comprises dynamic kernel shapes conducive to learning sentiment features of various scales to improve the model's Ewe text representation.

4.2.1. The multichannel 2D-CNN segment

This section ensures that all dimensions' input and output feature vectors remain unchanged, so the feature mapping group ($\hbar_{\upsilon}$) is pad to 0. Our study used dynamic kernel shapes (1 × 1, 3 × 3,} and 5 × 5) to extract sentiment features in different sentence lengths since we use DLSA.

The 2D-max-pooling is expressed as follows:

$$\hbar_{\upsilon}^{s} = \tanh(\hbar_{\upsilon} \times W_{s}). \tag{1}$$

$$\hbar_{\upsilon}^{s'} = \max(\hbar_{\upsilon}^{s}). \tag{2}$$

$$\hbar_{\upsilon}^{m} = \tanh(\hbar_{\upsilon} \times W_{m}). \tag{3}$$

$$\hbar_{\upsilon}^{m'} = \max(\hbar_{\upsilon}^{m}). \tag{4}$$

$$\hbar_{\upsilon}^{g} = \tanh(\hbar_{\upsilon} \times W_{g}). \tag{5}$$

$$\hbar_{\upsilon}^{g'} = \max(\hbar_{\upsilon}^{g}). \tag{6}$$

Note that $\hbar_{\upsilon}^{s} \in \mathbb{R}^{H\times I\times J_s}$, $\hbar_{\upsilon}^{m} \in \mathbb{R}^{H\times I\times J_m}$, and $\hbar_{\upsilon}^{g} \in \mathbb{R}^{H\times I\times J_g}$, are the 2D-convolutional output with 1 × 1, 3 × 3, and 5 × 5 kernel shapes. Furthermore, their corresponding dimensional output space is $\jmath_s$, $\jmath_m$, and $\jmath_g$ with max-pooling of $\hbar_{\upsilon}^{s'} \in \mathbb{R}^{H_s\times I_s\times J_s}$, $\hbar_{\upsilon}^{m'} \in \mathbb{R}^{H_m\times I_m\times J_m}$, and $\hbar_{\upsilon}^{g'} \in \mathbb{R}^{H_g\times I_g\times J_g}$, mapping with each convolution. The weight parameters are $H_s = I_s = H_m = I_m = H_g = I_g = {}^{H}/_{2} = {}^{I}/_{2}$ and $W^{s'}$, $W^{m'}$, $W^{g'}$, respectively. We encode the feature matrices $\hbar_{\upsilon}^{s'}$, $\hbar_{\upsilon}^{m'}$, and $\hbar_{\upsilon}^{g'}$ in a V-dimensional output space for concatenation ⊕ in their last stage. All encoded output features have a similar shape as follows:

$$\tilde{\hbar}_{\upsilon}^{s'} = ReLU\left(W^{s'} \times \hbar_{\upsilon}^{s} + \kappa_{s'}\right). \tag{7}$$

$$\tilde{\hbar}_{\upsilon}^{m'} = ReLU\left(W^{m'} \times \hbar_{\upsilon}^{m} + \kappa_{m'}\right). \tag{8}$$

$$\tilde{\hbar}_{\upsilon}^{g'} = ReLU\left(W^{g'} \times \hbar_{\upsilon}^{g} + \kappa_{g'}\right). \tag{9}$$

In our case, the $W^{s'}$, $W^{m'}$, $W^{g'}$ and $\kappa_{s'}$, $\kappa_{m'}$,$\kappa_{g'}$ represent the parameters of the output feature, such that $\tilde{\hbar}_{\upsilon}^{s'} \in \mathbb{R}^{H_s\times I_s\times V}$, $\tilde{\hbar}_{\upsilon}^{m'} \in \mathbb{R}^{H_m\times I_m\times V}$ and $\tilde{\hbar}_{\upsilon}^{g'} \in \mathbb{R}^{H_g\times I_g\times V}$ with ReLU as the activation function of the fully connected (FC) layer. Finally, we use an element-wise ⊕ mathematical expression to fuse all the feature matrices as:

$$\mathcal{F}_{\upsilon} = \left[\tilde{\hbar}_{\upsilon}^{s'} \oplus \tilde{\hbar}_{\upsilon}^{m'} \oplus \tilde{\hbar}_{\upsilon}^{g'}\right]. \tag{10}$$

With this, the multichannel network can get higher sentiment data than the standard CNN design with only one channel.

4.2.2. The attention-based BiLSTM (BiLSTM-Attn) segment

Our BiLSTM, based on the attention mechanism, allows us to examine the Ewe context in both directions and uses previous and future settings as described [40]. Two independent bidirectional hidden layers are used for this purpose, and their total output is then sent into the attention layer. This bidirectional network incorporates two hidden LSTM layers: forward L_n and backward L_m. The two layers investigate the feature sequences ρ from 1 to χ and backward. The BiLSTM study aims at determining the bidirectional annotations of words and ultimately summarizes the sentiment feature. Mathematically, our two bidirectional network layers are denoted as:

$$\overrightarrow{h_n} = L_n(\rho)_\chi, \chi \in [1, 100], \tag{11}$$

$$\overleftarrow{h_m} = L_m(\rho)_\chi, \chi \in [100, 1]. \tag{12}$$

The attention mechanism focuses on words that contribute most to a unique class of sentiment text. At this stage, the attention layer prioritizes significant sentiment traits and discards irrelevant ones but also pays extra attention to sentiment traits containing "unique" Ewe alphabetical letters. The densely deployed softmax layer processes the obtained features and generates the output. Specifically, we applied the outputs obtained from the BiLSTM layers to the attention mechanism, and words that contribute differently are given different weights. Attention and the BiLSTM model seek to compute the context vector denoted as:

$$a_\tau = \tanh(\mathrm{W} \cdot h_\tau + \kappa), \tag{13}$$

$$\eta_\tau = \frac{\exp(a_\tau^\pi \bullet a_\omega)}{\sum_\tau \exp(a_\tau^\pi \bullet a_\omega)}, \tag{14}$$

$$C = \sum_\tau^\pi \eta_\tau h_\tau, \tag{15}$$

where a_ω represents the context vector randomly initialized and subsequently learned during the training phase, and a_τ represents a concealed representation h_τ. To efficiently ascertain the value of a word a_τ, we compute a similarity check between a_ω and a_τ, while π is the overall time-steps in the input sequence. $\eta_\tau$is the computed weight at every step for each instance h_τ. C denotes the vector that summarizes all of the Ewe text in detail.

4.2.3. Ewe-based sentiment classifier

The model evaluates the sentiments using FC and activation layers. The sentiment output label is denoted as the distribution probability (D).

$$D = softmax(\tanh(\Upsilon^\mu \bullet W_1 + B_1)W_2 + B_2), \tag{16}$$

where Υ^μ is the feature output of the fused MC-2D-CNN and BiLSTM-Attn. The weight and bias parameters of our 2 FC layers are W_1, W_2 and B_1, B_2. This study uses a cross-entropy function as the fundamental loss function, and the class weights, including training and validation class weights, are calculated to maintain the overall training equilibrium.

5. Evaluation, Results, and Analysis

We show the efficacy of our proposed MC2D-CNN+BiLSTM-Attn by running an extensive experiment with n-fold cross-validation, where $n = 5$. We discuss the hyperparameter setting and analyze the results in this part. In this study, we use accuracy (acc.), precision (prec.), recall (rec.), error, and F1-score (F1s) as the model evaluation metrics for the sentiment analysis task as described in Verma [41]. We also compute the area under the receiver operating characteristic curve (AUC-ROC) curve for each benchmark method using our newly created EweSentiment dataset. Finally, we present a comprehensive result on all benchmark techniques to confirm our method's effectiveness.

5.1. Hyperparameter setting

We set our input channel to 60 and the compressed dimension's value to 900. For the first 2D-convolutional layer, we set 32 of the 1×1 filters, the second 2D-convolutional layer had 64 of the 3×3 filters, and the last 2D convolutional layer was set to 128 of 5×5 filters, respectively. The Adam optimizer trains the network at a LR of 0.002; the decay factor is set to 0.05, and a total training epoch of 250 for all the folds at a batch size of 16. To avoid system prediction error, the dropout and spatial dropout rates are adjusted to 0.5 and 0.15, respectively, and an L_1, L_2 regularization is used, as described [42]. We merge the BiLSTM method by a concatenation procedure, so the number of hidden units in LSTM is 128. For this study, we also use the generic cross-entropy loss function as our main loss function. Our experiments are performed on a single RTX 2080Ti GPU and implemented using the PyTorch-DL framework.

5.2. Results and analysis

Extensive experiments on the EweSentiment dataset with benchmark methods are presented in Table 2. Our proposed MC2D-CNN+BiLSTM-Attn method from Table 2 outperforms all three benchmark methods incorporating Glove, FastText, and Word2Vec models, respectively. Among these methods, BiLSTM incorporating FastText embedding as an input vector recorded the lowest in all performance metrics. The BiLSTM with Glove embedding had the highest accuracy of 0.727, precision of 0.709, recall of 0.691, error of 0.272, and F1s. of 0.700. This shows that the Glove embedding best represents the sentiment feature for the Ewe language in the case of the BiLSTM method. The overall performance of the BiLSTM with Word2Vec embedding is relatively good since we did not record any overfitting during training. The maximum fluctuation between Word2Vec and Glove embedding in acc., rec., and F1s. is 0.0149, 0.0236, and 0.0234, significantly lower than that of FastText.

Table 2
Comparing results of benchmark and proposed methods

Method	Embedding	Acc.	Prec.	Rec.	F1s.	Error
BiLSTM	**Glove**	**0.727**	**0.709**	**0.691**	**0.700**	**0.272**
	FastText	0.639	0.610	0.602	0.605	0.363
	Word2Vec	0.712	0.686	0.667	0.676	0.287
BiLSTM+Attn	Glove	0.815	0.791	0.802	0.790	0.184
	FastText	0.743	0.726	0.730	0.728	0.256
	Word2Vec	**0.848**	**0.827**	**0.826**	**0.827**	**0.151**
MC-CNN	Glove	0.851	0.836	0.806	0.821	0.148
	FastText	0.783	0.766	0.741	0.753	0.216
	Word2Vec	**0.896**	**0.871**	**0.861**	**0.866**	**0.103**
Proposed	Glove	0.903	0.895	0.902	0.898	0.096
(**Ours**)	FastText	0.886	0.849	0.818	0.833	0.113
	Word2Vec	**0.949**	**0.925**	**0.906**	**0.915**	**0.050**

In this study, we modify an attention framework and assimilate BiLSTM to create a new SOTA method termed Attn+BiLSTM. During our experiment, we trained the Attn+BiLSTM method from scratch using the newly created EweSentiment dataset to classify the predefined sentiment topics. Experimental results show that the attention mechanism influenced the models' performance. Comparatively, the Word2Vec model outperforms all other embeddings regarding acc., prec., rec., model error, and F1s.

It achieved 0.84838 accuracy, 0.82721 precision, 0.82697 recall, 0.15162 model error, and 0.82708 F1s, respectively. In the case of the Glove model, an acc. of 0.81573, a prec. of 0.79170, a rec. of 0.80265, an F1s. of 0.79071, and an error of 0.18427 were achieved. The overall performance for the FastText was significantly lower, with a maximum fluctuation in accuracy and precision of 0.105 for Word2Vec and 0.0725 for the Glove, respectively.

For the MC-CNN with a one-dimensional layer, our Ewe-based Word2Vec embedding was superior to Glove and FastText embedding in terms of acc., prec., rec., F1s., and a tiny model error rate (misclassification) (see Table 2). Also, the MC-CNN method considerably performs better than BiLSTM and Attn +BiLSTM in terms of all performance metrics. Using the Word2Vec embedding as an input vector for the MC-CNN method, an acc. of 0.8965, a prec. score of 0.8718, rec. of 0.8618, and an F1s. of 0.8668 with 0.1034 misclassifications (error) were obtained.

Subsequently, our proposed MC2D-CNN+BiLSTM-Attn method outperformed BiLSTM, Attn+BiLSTM, and MC-CNN methods regarding precision, recall, accuracy, and F1-score (see Table 2). However, compared with embedding models, the total improvements on FastText are smaller than those on Glove and Word2Vec. The key reason is that, due to the complex nature of our proposed method, it can completely obtain textual sentiment information compared with other methods, hence acquiring additional sentiment data in most samples, thereby obtaining better performance and balancing the effects of overfitting. The proposed method utilizing Word2Vec embedding as an input tensor recorded the best performance scores of 0.9493 in accuracy, 0.9257 in precision, a recall of 0.9063, and an F1-score of 0.9159 with a tiny error of 0.0506. In machine learning, evaluating a model's performance is key to its robustness and stability. With this, the proposed novel method performs better on the EweSentiment datasets than the existing benchmark methods. Our strategy fully exploits the merits of CNN, attention, and BiLSTM mechanisms. Moreover, our approach can extract "unique" and high-level Ewe context utilizing a multichannel framework with different kernel sizes, allowing us to completely analyze and classify sentiment data in the Ewe language. Additionally, the attention model employed plays a unique role by critically capturing Ewe sentiment features in the BiLSTM throughout the entire procedure, efficiently filtering out exact Ewe features. In terms of accuracy, recall, precision, and F1-scores evaluation metrics, the proposed method outperforms compared benchmark methods, proving its stability and robustness in classifying the exact Ewe sentiment features.

We further present and visualize the performance of the proposed method in solving the Ewe-based sentiment analysis problem by further computing the AUC-ROC (Table 3). We illustrate in Figure 3 the difference in the AUC-ROC score between the word embedding model and the EweSentiment dataset. The proposed method highlighted the highest AUC-ROC values for Glove as 0.9354, FastText as 0.9179, and Word2Vec as 0.9680, respectively.

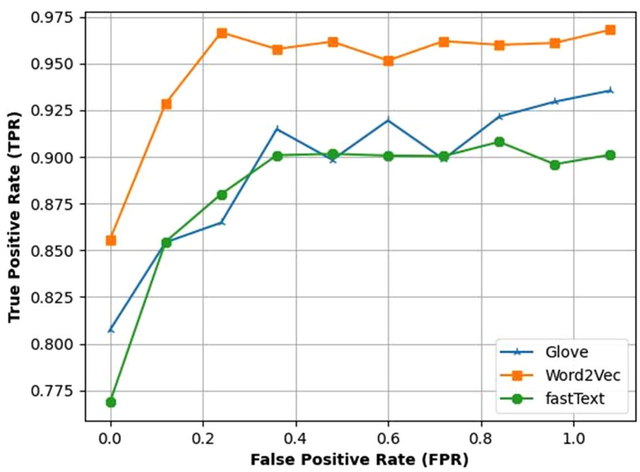
5.3. Ablation study

In this part, we further ascertain the stability and robustness of our proposed MC2D-CNN+BiLSTM-Attn method. We discuss and analyze the effects and influence of exploring different LRs on the proposed method. We also present an in-depth error analysis of classifying the exact Ewe sentiment features from the newly developed EweSentiment dataset.

Table 3
Classification results on AUC-ROC

Method	Embedding	AUC-ROC
BiLSTM	**Glove**	0.8777
	FastText	0.7596
	Word2Vec	0.8668
BiLSTM+Attn	Glove	0.8714
	FastText	0.7781
	Word2Vec	0.8917
MC-CNN	Glove	0.9295
	FastText	0.8873
	Word2Vec	0.9419
Proposed (Ours)	Glove	0.9354
	FastText	0.9179
	Word2Vec	0.9680

Figure 3
AUC-ROC score



5.3.1. *Effect of different LRs*

We survey the impact of different LRs of the optimization procedure employed on the proposed MC2D-CNN+BiLSTM-Attn method. During our experiment, we set a succession of LRs (0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, and 0.008) for each embedding. An extensive simulation indicates that, at an LR of 0.002, the proposed method performs better with Word2Vec, Glove, and FastText embedding than other LRs. For this part, the authors focused only on the Word2Vec embedding with the proposed method because its results outperformed all other embeddings at the above-mentioned LRs (Table 4).

From Table 4, we observed that at a 0.001 LR, the proposed method had a fair performance score with an accuracy of 0.9209, a precision of 0.9333, a recall of 0.9032, and an F1-score of 0.91802, respectively. Ultimately, the model's efficiency was superior at a LR of 0.002, with F1-score, recall, precision, and accuracy stretching to 0.91593, 0.9063, 0.9257, and 0.9493, respectively. As the LR values increased, MC2D-CNN+BiLSTM-Attn performance decreased (Figure 4). When the LR was set to 0.008, the accuracy, precision, and F1-score declined to 0.8002, 0.7899, 0.7697, and 0.7797, indicating the method's worst performance. This decline also occurred in all other embeddings, showing that the best LR for the proposed model using the

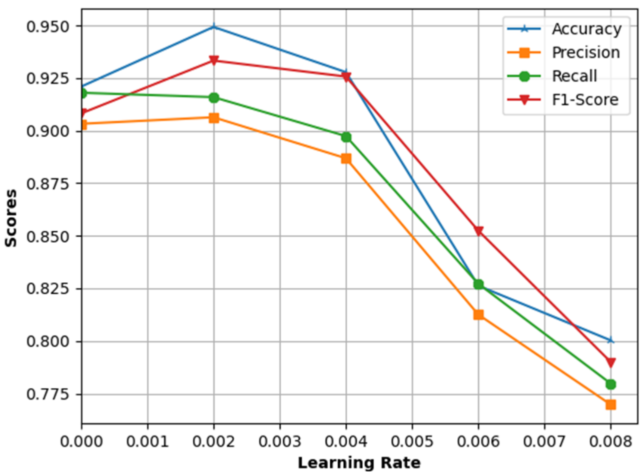
Table 4
Effects of different learning rates

Method	Embedding	Acc.	Prec.	Rec.	F1s.
0.001	Glove	**0.9007**	**0.8814**	**0.8951**	**0.8882**
	FastText	0.8787	0.8576	0.8265	0.8418
	Word2Vec	0.9209	0.9333	0.9032	0.9180
0.002	Glove	0.9031	0.8951	0.9020	0.8985
	FastText	0.8865	0.8494	0.8181	0.8334
	Word2Vec	**0.9493**	**0.9257**	**0.9063**	**0.9159**
0.004	Glove	0.9218	0.8614	0.8738	0.8676
	FastText	0.8511	0.8200	0.7956	0.8076
	Word2Vec	0.9277	0.9081	0.8868	0.8973
0.006	Glove	0.8718	0.8623	0.8448	0.8534
	FastText	0.8129	0.7926	0.7899	0.7913
	Word2Vec	0.8862	0.8424	0.8125	0.8271
0.008	Glove	0.7200	0.6713	0.6204	0.6449
	FastText	0.6353	0.6281	0.6006	0.6141
	Word2Vec	0.8002	0.7899	0.7697	0.7797

EweSentiment dataset is 0.002. In the difference in interval context, the performance trend between 0.001 and 0.002 shows a relative increase but records a continuous decline from 0.003 to 0.008 in all four-evaluation metrics under Glove and FastText embedding. Though its predictive accuracy fluctuated between 0.001 and 0.003, its fluctuation strength is quite minor, showing that the model performance was mostly unaffected by the LR at that time and was comparatively steady. The model's efficiency steadily declined when the LR surpassed 0.003. The weakest scores for each model assessment parameter were obtained when the LR reached 0.008, showing that once it was above 0.003, the model started to suffer from its negative effects, which worsened as the LR increased. These detrimental effects caused the model's capacity to extract and use sentiment characteristics to deteriorate, which further impacted the model's functionality.

This study validates the proposed methods' stability when setting different LRs. With experience in training, we realize that the proposed method performs worst when the LR is set from 0.005 to 0.008. The method's overall stability was optimal when the LR was 0.002. Figure 4 further shows that when the LR was 0.001, the proposed method stability exhibited fair stability but recorded a strong ascending score of 0.002, and then, surprisingly,

Figure 4
Results on different learning rates



a descending pattern was reached from 0.003 to 0.005, and then at 0.006, 0.007, and 0.008 respectively. This demonstrates that while there were variations throughout this time, the method's stability had minimal impact and was satisfactory. This fluctuation diminished as the LR exceeded 0.003 and was replaced by a constant upward trend, indicating the methods' stability changed significantly as the LR exceeded 0.003 and unstable factors surged, which is not opportune for the method learning all Ewe sentiment features.

The in-depth study above shows that LRs impact the model's performance and stability. Choosing the right LR is essential to enhance the model's performance and preserve its stability. Different LRs affect the model most since a slow LR makes it hard for the model to learn the text's characteristics during training.

5.3.2. Error analysis

We examine possible misclassified sentiment features with the proposed method. In particular, certain cases of error prediction in each 5-fold are chosen randomly from the validation set. Since this is the first-ever sentiment analysis study in the Ewe language, the following explanations are given for the classification errors based on the three embeddings.

Firstly, the complexity of our method makes it difficult to predict Ewe's questioning tone and text with contrasting sentiment words. For example, "Wo fofo gbea woƒe sukufewo xexe le woƒe nuwɔna gbegbl ta" (Their dad refuses to pay their school fees as a result of their misconduct.) The actual sentiment of this phrase is "anger," yet the model predicted "sad." Also, another sentence: "Vudo! eƒe sisi ƒe ablaɖeɖe gakpɔtɔ nye vlododo! Enye aƒetɔ ƒe nuwɔna eye wòfa." (Well! his tempo of escape remains despicable! It's aristocratic and cool.). The actual sentiment of this phrase is ''sad,'' but the model prediction is "surprise." In this case, we realize that it was easy for our method to acquire sentiment bias from texts like "cool," "misconduct, refuse, and "aristocratic," which are significant sentiment texts, yet the questioning tone induces it to convey contrary sentiments. As a result, it is difficult to create an accurate prediction. Secondly, extremely concealed sentiments cannot be detected in the Ewe language. For sentences like "Ètrɔ tso esime mèwɔ ɖe susu kple nukpɔsusu aɖewo dzi o." (You have changed since you did not follow certain thoughts and views.), the optimal sentiment is ''surprise'', yet it is ''anger'', and "Amedzroa mete ŋu ɖu nu le kplɔ̃ɖoƒea o le zimenɔla ƒe nya gbegbl a ta." (The guest couldn't eat at the party due to the chairman's bad comment.) the real sentiment is "anger," but the model predicted "surprise."

Lastly, we realize that the length of the sentence influences the final prediction. If the Ewe phrase is long and the words that show the main emotion are attached, our proposed model could not find a valid sentiment topic. This would lead to misclassification or the wrong classification.

6. Discussion and Implication

In this study, we developed an EweSentiment dataset. The developed dataset consists of individual comments gathered on various social media platforms on five sentiment topics. Based on these five basic sentiment topics, we generated an annotated dataset of approximately 38K comments comprising 4315 total annotations. We aimed to create three different word embeddings based on the EweSentiment dataset. We created Glove, Word2Vec, and FastText. These embeddings were input vectors that fed into the proposed methods to determine the exact sentiment features. Experiments indicated that Word2Vec embedding outperformed the Glove and FastText in incorporating the proposed method. Finally, we aimed to capture exact sentiment features using the proposed method. The experiment results showed that the proposed method did better than the compared methods at getting the exact Ewe-based features of individual sentiment.

As social media grows in popularity, the huge volume of user-generated information provides significant insights to help us better comprehend public sentiment. Since social media users are increasingly transitioning from publishing simple textual messages to expressing themselves in detailed documents, the rich information they provide is valuable in many NLP applications. To this end, our proposed MC2D-CNN+BiLSTM-Attn method can robustly integrate Ewe textual information to derive the exact Ewe representation for DLSA, which proved successful on the EweSentiment dataset. Adequately, the method captures semantic information and detailed sentiment features formed from Ewe alphabetical letters (ɖ, ʋ, ƒ, í, ŋ, ɔ, ɖ, ɔ̃, ã, ɛ, ʋ, ê, etc.), which are not part of the English alphabet. Using our method, researchers, experts, and users may rapidly assess the extracted feature terms and accompanying sentiment polarity from a huge scale of social media comments in other African languages, allowing them to better understand public opinions from various platforms.

This study contributes to society by enabling businesses, governments, organizations, and individuals to gain valuable insights from text data, make informed decisions, and respond to public sentiment effectively. It enhances customer experiences, drives innovation, and aids in addressing societal challenges. Its applications span various industries and domains, making it a valuable tool in the modern world. Theoretically, this study adds to the literature on URLs that have not been studied, especially in West Africa. Also, the study adds to the body of knowledge on using multichannel 2D-CNN and BiLSTM-attention in extracting sentiment features from LRLs such as the Ewe Language. Also, researchers from developing countries, like Ghana, Benin, Togo, Nigeria, and Niger-Congo, can use the EweSentiment dataset to research sentiment analysis in the future.

7. Conclusion

We collected and preprocessed an Ewe-based document-level sentiment dataset named EweSentiment, which consists of comments from various social media platforms on five sentiment topics. These comments were translated into the Ewe language and extensively checked by five Ewe tutors in Ghana to ensure exact and concise sentiment representation. We propose a multichannel 2D-CNN with an attention-based Bi-LSTM method to solve the sentiment problem associated with the Ewe language efficiently. Specifically, we construct three Ewe-based word embeddings, such as Glove, Word2Vec, and FastText, for exploiting sentiment representation. We evaluate the performance of the embeddings in classifying Ewe sentiment features with BiLSTM, BiLSTM+Attn., MC-CNN, and the proposed MC2D-CNN+BiLSTM-Attn method. Results suggest that Word2Vec is superior to the proposed method for learning Ewe sentiment features. Also, we visualize and analyze models' robustness and stability under different LRs. We compare and analyze factual error instances in predicting the exact Ewe sentiment. The empirical evidence shows that the proposed MC2D-CNN +BiLSTM-Attn method performs better at classifying the Ewe sentiment text than the three SOTA methods. In conclusion, the Ewe Sentiment analysis contributes to society by providing valuable insights from textual data, facilitating data-driven decision-making, enhancing communication, and addressing various challenges across sectors, including politics, education, health, and the business industry. Its applications span diverse industries, making it a valuable tool for understanding and responding to public sentiment in the modern world.

In our future study, we aim to report on extra embedding procedures, such as pretrained word embeddings with different dimensions and XLNET, and identify the most efficient procedures for solving large-scale word embedding issues, which are currently time-consuming in the low-resourced NLP research field.

Funding Support

This work was partially supported by the National Key Research and Development Program of China under Grant 2018AAA0100202 and the National Science Foundation of China under Grant 61976043.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Author Contribution Statement

Victor Kwaku Agbesi: Conceptualization, Methodology, Software, Formal analysis, Writing – original draft. **Wenyu Chen:** Supervision, Project administration. **Chiagoziem C. Ukwuoma:** Formal analysis, Writing – review & editing, Visualization, Project administration. **Noble A. Kuadey:** Investigation, Writing – review & editing, Visualization. **Collinson Colin M. Agbesi:** Investigation, Writing – review & editing. **Chukwuebuka J. Ejiyi:** Validation, Data curation. **Emmanuel S. A. Gyarteng:** Validation, Data curation. **Gladys W. Muoka:** Validation, Visualization. **Anthony M. Kuadey:** Software, Data curation.

References

[1] Agbesi, V. K., Chen, W., Gizaw, S. M., Ukwuoma, C. C., Ameneshewa, A. S., & Ejiyi, C. J. (2023). Attention based BiGRU-2DCNN with hunger game search technique for low-resource document-level sentiment classification. In *Proceedings of the 2023 5th International Conference on Image Processing and Machine Vision*, 48–54. https://doi.org/10.1145/3582177.3582186

[2] Shirsat, A. R., González, A. F., & May, J. J. (2022). Proposing a model of social media user interaction with fake news. *Journal of Information, Communication and Ethics in Society*, *20*(1), 134–149. https://doi.org/10.1108/JICES-10-2020-0104

[3] Nagamanjula, R., & Pethalakshmi, A. (2020). A novel framework based on bi-objective optimization and LAN^{2}FIS

for Twitter sentiment analysis. *Social Network Analysis and Mining*, *10*(1), 34. https://doi.org/10.1007/s13278-020-00648-5

[4] Naz, S., Sharan, A., & Malik, N. (2018). Sentiment classification on Twitter data using support vector machine. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence,* 676–679. https://doi.org/10.1109/WI.2018.00-13

[5] Bibi, R., Qamar, U., Ansar, M., & Shaheen, A. (2019). Sentiment analysis for Urdu news tweets using decision tree. In *2019 IEEE 17th International Conference on Software Engineering Research, Management and Application,* 66–70. https://doi.org/10.1109/SERA.2019.8886788

[6] Jaha, A., Satya Deepthi, N., Suryakanth, G., & Surya Kala Eswari, G. (2020). Text sentiment analysis using Naïve Baye's classifier. *International Journal of Computer Trends and Technology*, *64*(4), 261–265.

[7] Zhang, Z., Zou, Y., & Gan, C. (2018). Textual sentiment analysis via three different attention convolutional neural networks and cross-modality consistent regression. *Neurocomputing*, *275*, 1407–1415. https://doi.org/10.1016/j.neucom.2017.09.080

[8] Gan, C., Wang, L., & Zhang, Z. (2020). Multi-entity sentiment analysis using self-attention based hierarchical dilated convolutional neural network. *Future Generation Computer Systems*, *112*, 116–125. https://doi.org/10.1016/j.future.2020.05.022

[9] Abdi, A., Shamsuddin, S. M., Hasan, S., & Piran, J. (2019). Deep learning-based sentiment classification of evaluative text based on multi-feature fusion. *Information Processing & Management*, *56*(4), 1245–1259. https://doi.org/10.1016/j.ipm.2019.02.018

[10] Zhang, Z., Wang, L., Zou, Y., & Gan, C. (2018). The optimally designed dynamic memory networks for targeted sentiment classification. *Neurocomputing*, *309*, 36–45. https://doi.org/10.1016/j.neucom.2018.04.068

[11] Basiri, M. E., Nemati, S., Abdar, M., Cambria, E., & Acharya, U. R. (2021). ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. *Future Generation Computer Systems*, *115*, 279–294. https://doi.org/10.1016/j.future.2020.08.005

[12] Vilares, D., Peng, H., Satapathy, R., & Cambria, E. (2018). BabelSenticNet: A commonsense reasoning framework for multilingual sentiment analysis. In *2018 IEEE Symposium Series on Computational Intelligence,* 1292–1298. https://doi.org/10.1109/SSCI.2018.8628718

[13] Mahmood, Z., Safder, I., Nawab, R. M. A., Bukhari, F., Nawaz, R., Alfakeeh, A. S., . . . , & Hassan, S. U. (2020). Deep sentiments in Roman Urdu text using recurrent convolutional neural network model. *Information Processing & Management*, *57*(4), 102233. https://doi.org/10.1016/j.ipm.2020.102233

[14] Sharf, Z., & Rahman, S. U. (2018). Performing natural language processing on Roman Urdu datasets. *International Journal of Computer Science and Network Security*, *18*(1), 141–148.

[15] Mehmood, K., Essam, D., Shafi, K., & Malik, M. K. (2020). An unsupervised lexical normalization for Roman Hindi and Urdu sentiment analysis. *Information Processing & Management*, *57*(6), 102368. https://doi.org/10.1016/j.ipm.2020.102368

[16] Sarkar, K. (2019). Sentiment polarity detection in Bengali tweets using deep convolutional neural networks. *Journal of Intelligent Systems*, *28*(3), 377–386. https://doi.org/10.1515/jisys-2017-0418

[17] Emon, E. A., Rahman, S., Banarjee, J., Das, A. K., & Mittra, T. (2019). A deep learning approach to detect abusive Bengali text. In *2019 7th International Conference on Smart Computing and Communications,* 1–5. https://doi.org/10.1109/ICSCC.2019.8843606

[18] Irtiza Trinto, N., & Eunus Ali, M. (2018). Detecting multilabel sentiment and emotions from Bangla Youtube comments. In *2018 International Conference on Bangla Speech and Language Processing*, 1–6. https://doi.org/10.1109/ICBSLP.2018.8554875

[19] Mahmud, M. S., Islam, M. T., Bonny, A. J., Shorna, R. K., Omi, J. H., & Rahman, M. S. (2022). Deep learning based sentiment analysis from Bangla text using glove word embedding along with convolutional neural network. In *2022 13th International Conference on Computing Communication and Networking Technologies*, 1–6. https://doi.org/10.1109/ICCCNT54827.2022.9984392

[20] Tasnim, F., Habiba, S. U., Nafisa, N., & Ahmed, A. (2022). Depressive Bangla text detection from social media post using different data mining techniques. In *Computational Intelligence in Machine Learning: Select Proceedings of ICCIML 2021*, 237–247. https://doi.org/10.1007/978-981-16-8484-5_21

[21] Sarkar, K. (2020). Sentiment analysis of Bengali tweets using deep learning. In *Computational Intelligence in Data Science: Third IFIP TC 12 International Conference*, *578*, 71–84. https://doi.org/10.1007/978-3-030-63467-4_6

[22] Azmin, S., & Dhar, K. (2019). Emotion detection from Bangla text corpus using Naïve Bayes classifier. In *2019 4th International Conference on Electrical Information and Communication Technology*, 1–5. https://doi.org/10.1109/EICT48899.2019.9068797

[23] Sharmin, S., & Chakma, D. (2021). Attention-based convolutional neural network for Bangla sentiment analysis. *AI & Society*, *36*(1), 381–396. https://doi.org/10.1007/s00146-020-01011-0

[24] Rahman, M., Haque, S., & Saurav, Z. R. (2020). Identifying and categorizing opinions expressed in Bangla sentences using deep learning technique. *International Journal of Computer Applications*, *176*(17), 13–17. https://doi.org/10.5120/ijca2020920119

[25] Islam, K. I., Islam, M. S., & Amin, M. R. (2020). Sentiment analysis in Bengali via transfer learning using multi-lingual BERT. In *23rd International Conference on Computer and Information Technology*, 1–5. https://doi.org/10.1109/ICCIT51783.2020.9392653

[26] Smetanin, S., & Komarov, M. (2021). Deep transfer learning baselines for sentiment analysis in Russian. *Information Processing & Management*, *58*(3), 102484. https://doi.org/10.1016/j.ipm.2020.102484

[27] Agbesi, V. K., Chen, W., Kuadey, N. A., & Maale, G. T. (2022). Multi-topic categorization in a low-resource ewe language: A modern transformer approach. In *2022 7th International Conference on Computer and Communication Systems*, 42–45. https://doi.org/10.1109/ICCCS55155.2022.9846372

[28] Fouad, M. M., Mahany, A., Aljohani, N., Abbasi, R. A., & Hassan, S. U. (2020). ArWordVec: Efficient word embedding models for Arabic tweets. *Soft Computing*, *24*(11), 8061–8068. https://doi.org/10.1007/s00500-019-04153-6

[29] Rizkallah, S., Atiya, A. F., Shaheen, S., & Mahgoub, H. E. D. (2023). ArSphere: Arabic word vectors embedded in a polar

sphere. *International Journal of Speech Technology*, *26*(1), 95–111. https://doi.org/10.1007/s10772-022-09966-9
[30] Alzanin, S. M., Azmi, A. M., & Aboalsamh, H. A. (2022). Short text classification for Arabic social media tweets. *Journal of King Saud University - Computer and Information Sciences*, *34*(9), 6595–6604. https://doi.org/10.1016/j.jksuci.2022.03.020
[31] Mittal, N., & Agarwal, B. (2013). Sentiment analysis of Hindi review based on negation and discourse relation. In *International Joint Conference on Natural Language Processing*, 45–50.
[32] Al-Amin, M., Islam, M. S., & Das Uzzal, S. (2017). Sentiment analysis of Bengali comments with Word2Vec and sentiment information of words. In *2017 International Conference on Electrical, Computer and Communication Engineering*, 186–190. https://doi.org/10.1109/ECACE.2017.7912903
[33] Agbesi, V. K., Chen, W., Ameneshewa, A. S., Odame, E., Dumor, K., & Browne, J. A. (2023). Efficient adaptive convolutional model based on label embedding for text classification using low resource languages. In *Proceedings of the 2023 7th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*, 144–151. https://doi.org/10.1145/3596947.3596962
[34] Nandwani, P., & Verma, R. (2021). A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining*, *11*(1), 81. https://doi.org/10.1007/s13278-021-00776-6
[35] Hameed, Z., & Garcia-Zapirain, B. (2020). Sentiment classification using a single-layered BiLSTM model. *IEEE Access*, *8*, 73992–74001. https://doi.org/10.1109/ACCESS.2020.2988550
[36] Samiul Salehin, S. M., Miah, R., & Saiful Islam, M. (2020). A comparative sentiment analysis on Bengali Facebook posts. In *Proceedings of the International Conference on Computing Advancements*, 21. https://doi.org/10.1145/3377049.3377078
[37] Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing,* 1532–1543. https://doi.org/10.3115/v1/d14-1162
[38] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://doi.org/10.1162/tacl_a_00051
[39] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv Preprint: 1301.3781*.
[40] Anki, P., & Bustamam, A. (2021). Measuring the accuracy of LSTM and BiLSTM models in the application of artificial intelligence by applying chatbot programme. *Indonesian Journal of Electrical Engineering and Computer Science*, *23*(1), 197–205. https://doi.org/10.11591/ijeecs.v23.i1.pp197-205
[41] Verma, S. (2022). Sentiment analysis of public services for smart society: Literature review and future research directions. *Government Information Quarterly*, *39*(3), 101708. https://doi.org/10.1016/j.giq.2022.101708
[42] Li, C., Wang, B., Pavlu, V., & Aslam, J. A. (2016). An empirical study of skip-gram features and regularization for learning on sentiment analysis. In *Advances in Information Retrieval: 38th European Conference on IR Research,* 72–87. https://doi.org/10.1007/978-3-319-30671-1_6

How to Cite: Agbesi, V. K., Chen, W., Ukwuoma, C. C., Kuadey, N. A., Agbesi, C. C. M., Ejiyi, C. J., . . . , & Kuadey, A. M. (2025). Multichannel 2D-CNN Attention-Based BiLSTM Method for Low-Resource Ewe Sentiment Analysis. *Journal of Data Science and Intelligent Systems*, *3*(1), 67–77. https://doi.org/10.47852/bonviewJDSIS32021512