Journal of Business, Communication & Technology

Attention-Driven Bidirectional LSTM Neural Network for Afaan Oromo Next Word Generation

Bekan Kitaw Mekonen[1]*
[1] *Jimma University, Ethiopia*

Abstract Effective communication through digital platforms often faces issues like misspellings and inefficient typing. A next-word prediction system that suggests probable words can significantly enhance sentence construction, especially for Afaan Oromo - a Cushitic language spoken by over 41.7 million people in Ethiopia. Despite its importance as the official language of Oromia and its complex linguistic features, Afaan Oromo lacks advanced digital tools. This study evaluates various deep learning models, including Long Short Term Memory (LSTM), Attention-based LSTM, Bidirectional LSTM (Bi-LSTM), Attention-based Bi-LSTM, and Recurrent Neural Network (RNN), to determine the most accurate model for Afaan Oromo next word generation. Our methodology involves developing and benchmarking these models using a comprehensive dataset of 201,538 words sourced from various media, academic literature, and religious texts. The Attention-driven Bi-LSTM model emerged as the most effective, achieving an accuracy of 95.0% and a low loss value of 0.27. These findings highlight the potential of the Attention-driven Bi-LSTM model to improve the next word generation for Afaan Oromo texts. This advancement addresses specific linguistic challenges and enhances the overall digital interaction experience for Afaan Oromo speakers.

Keywords: *Word generation, Deep Learning, Attention Driven, Bi-LSTM, Communication*

***Corresponding Author:**
Bekan Kitaw Mekonen
bekan.mekonen@ju.edu.et

Received: September 2024
Revised: November 2024
Accepted: December 2024
Published: December 2024

1. Introduction

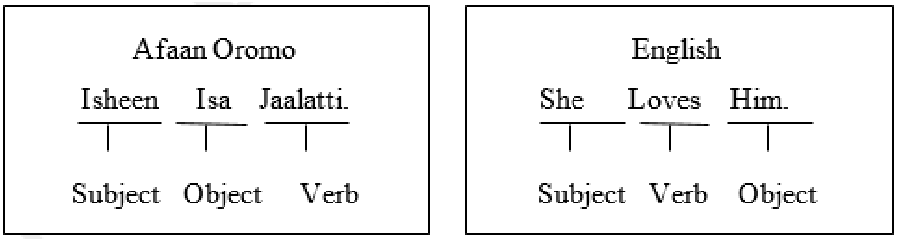
Communication among humans, primarily through natural language, remains an effective method for sharing information. Natural Language Processing (NLP), a branch of Artificial Intelligence (Khurana et al., 2022), plays a significant role in advancing language development, particularly in this digital age. However, a substantial portion of the global population, especially those in low-resource communities (like the majority of Africans), face difficulty accessing AI technologies. Afaan Oromo, a widely spoken Cushitic language in Ethiopia, is one such under-resourced language (Walga, 2021). It is officially recognized as the working language of the Oromia regional state and has been proposed as one of the four

http://doi.org/10.56632/bct.2025.4104

working languages of the Ethiopian federal government. With a speaker population exceeding 40 million, Afaan Oromo is the most widely spoken Cushitic language in Africa and has the highest number of native speakers in Ethiopia (Garoma, 2024).

Despite its cultural significance and widespread use, the field of NLP for Afaan Oromo is significantly underdeveloped, which highlights the necessity of conducting this research. The lack of robust language processing tools hinders its potential in various domains, including education and technology. Factors such as the scarcity of annotated linguistic data, limited funding for linguistic research, and the absence of effective language tools all contribute to the language's under-resourced status, affecting its preservation and development. Furthermore, variation in Afaan Oromo orthography, especially among non-native speakers, exacerbates these challenges. The lack of a next-word prediction system (Liu & Sun, 2023) leads to substantial semantic ambiguities. That is, a subtle difference in word choice can drastically alter the meaning of a sentence. For example, in Afaan Oromo, the correct use of "*Gadaa*" (system) versus "*Gaddaa*" (mourning) in the statement "Barnootni Keenya har'aa waa'ee sirna Gadaati." meaning 'Our lesson today is about the Gada system.' and the statement "Barnootni Keenya har'aa waa'ee sirna Gaddaati." meaning 'Our lesson today is about the ritual of Mourning.' highlights the potential for misinterpretation without appropriate language tools.

Afaan Oromo uses '*Qubee*' script, a Latin-based alphabet comprising 32 characters: five vowels ('a', 'e', 'i', 'o', 'u'), also known as *dubbachiiftuu*, and 27 consonants (including paired consonants (*qubee dachaa*) like 'ch', 'dh', 'ny', 'sh', 'ts', 'ph'), also called *dubbifamaa* (Gemeda, 2023). Similar to English, Afaan Oromo distinguishes between uppercase and lowercase letters. Vowels function as independent sounds, and both short (*sagalee gabaabaa*, as in '*nama*', meaning human) and long vowel (*sagalee dheeraa*, as in '*diimaa*', meaning red) variations exist. Moreover, sentence boundaries (known as '*hima*') in Afaan Oromo are typically indicated with punctuation like periods (.), question marks (?), and exclamation marks (!) indicating sentence ends, akin to English. Additionally, word segmentation (known as '*jecha*') in Afaan Oromo, like many Latin-based languages, relies on spaces to separate words. The language also exhibits a rich morphological system comparable to other African and Ethiopian languages, adhering to a Subject-Object-Verb (SOV) order (Tesema & Tamirat, 2017), unlike the Subject-Verb-Object (SVO) structure of English. Adjective placement differs, with Afaan Oromo adjectives typically following nouns, whereas English adjectives usually precede them. For instance,



Hence, this study aims to develop a next-sequence generation system to enhance the accuracy and clarity of Afaan Oromo texts by providing attention-driven word suggestions. Therefore, the primary goals of this study are to:

- Conduct a thorough analysis of Afaan Oromo's linguistic structure and morphology.
- Develop a substantial corpus of Afaan Oromo text.
- Design and implement a word sequence generation model for Afaan Oromo.
- Evaluate the performance of the developed model using the collected dataset.

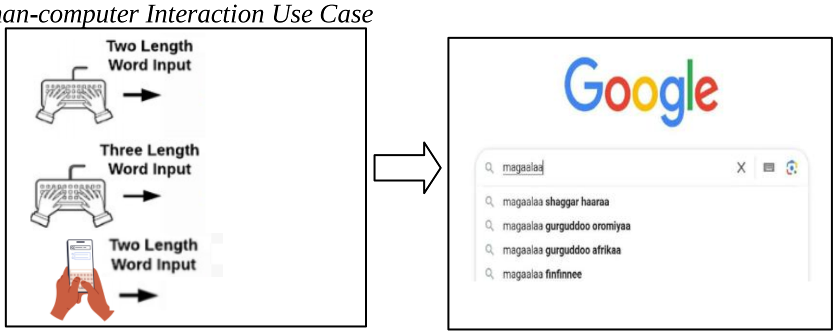
Moreover, the research questions guiding this study are:

- Can deep learning models be used for the next-word generation in Afaan Oromo?
- How does the choice of the Deep Learning model impact the next-word generation performance?
- How does the integration of attention mechanisms with Bidirectional LSTM networks improve the model's ability to capture contextual dependencies in Afaan Oromo text?

2. Theoretical Framework

Next-word generation is a fundamental task in NLP and has been extensively researched using various linguistic models. Language modeling, also known as linguistic modeling, started during the 1980s when the primary models of some importance were developed (Iqbal & Qureshi, 2020). These first models were designed for written text, and from that moment onward, the models have been modified and enhanced to incorporate spoken languages. Early methods relied on the use of n-gram models (Bickel et al., 2005), which predict the subsequent word based on the preceding n-1 words. These models, though simple and efficient, face challenges such as sparse data issues and the inability to capture long-range dependencies. Nowadays, language models are integral in accelerating communication between individuals and in human-computer interactions, including applications like smart keyboards, email response suggestions, auto-correction for spelling, and virtual assistants.

Figure 1
Accelerated Human-computer Interaction Use Case



However, these applications face particular challenges when applied to languages such as Afaan Oromo, which has unique linguistic features that complicate the next-word generation. For example, Afaan Oromo's Subject-Object-Verb (SOV) structure contrasts with the more commonly modeled Subject-Verb-Object (SVO) structure of languages like English. This difference significantly affects word order prediction and introduces ambiguity in the sequence of word predictions. Moreover, the language's complex morphology, including rich inflectional and derivational processes, poses additional challenges for predicting the correct noun and verb forms. The interplay between these morphological features and word order adds layers of complexity that models must account for, making it harder for traditional models to predict the next word accurately. These challenges highlight the need for models that are specifically tailored to address the linguistic properties of morphologically rich languages like Afaan Oromo.

Several studies have investigated next-word generation and sentence completion in various languages, applying different approaches. While traditional NLP models, such as n-gram and support vector machines (SVMs), have been utilized for next-word generation, they often struggle with limitations that deep learning approaches effectively address. N-gram models, for example, are limited by context window size and are unable to model long-range dependencies, while SVMs face difficulty in capturing the intricacies of under-resourced language. Hence, with the emergence of deep learning, RNNs have been utilized for next-word generation due to their capability to model sequential data. For instance, Patil et al. (2024) suggested that RNNs maintain a hidden state that captures information from previous time steps. Besides, the power of RNN for next-word prediction in Assamese phonetic transcription is also studied (Barman & Boruah, 2018). However, traditional RNNs still struggle to learn long-range dependencies due to issues like vanishing gradients.

To overcome the limitations of these traditional RNNs, Long Short-Term Memory (LSTM) networks were introduced. As discussed by Ambulgekar et al. (2021), LSTM neural networks are used to predict the next word in a sequence, utilizing character-level prediction. They trained the model on a Nietzsche text corpus and showed a significant improvement over traditional n-gram approaches due to LSTM's capacity to capture long-term dependencies. Despite attaining a moderate accuracy of ~56%, the study highlights the potential of LSTM for improving text prediction applications. Sumathy et al. (2023)

utilize LSTM models for subsequent word prediction using a corpus obtained through web scraping of Indonesian data, achieving an improved accuracy of 75% over 20 epochs. The study illustrates the superiority of LSTM models in handling sequential data and enhancing prediction accuracy compared to traditional RNNs and federated text models. Wangchuk et al. (2023) discuss the challenges of digitizing the Dzongkha language and propose a Bi-LSTM model for Dzongkha word prediction. The model achieved an accuracy of 73.89% with a loss of 1.0722. The study aims to reduce keystrokes and make Dzongkha typing quicker and more efficient to bridge the digital divide in Bhutan.

The comparative analysis of BI-LSTM and LSTM was conducted in the works by Rathee and Yede (2023) and Sharma et al. (2019), utilizing the English-Hindi corpus as a dataset. From their analysis, they concluded that Bi-LSTM performed better than the basic LSTM, with accuracies of 81.07% and 59.46%, respectively. The study by Ganai and Khursheed (2019) explores the application of RNN and LSTM models for predicting the next word in language modeling, highlighting their use in structured document retrieval. The authors propose a tree-based generative language model for ranking documents and parts, with well-defined language models at each node within the document hierarchy. The work also demonstrates the limitations of N-gram models and the advantages of RNNs in language modeling. The LSTM and GRU models predict disaster events on Twitter data, and comparing their performance with and without word embedding was explained (Bhuvaneswari et al., 2019). Accordingly, the Bidirectional GRU achieved higher accuracy in contrast with the other. Moreover, hybrid approaches have also been proposed in numerous studies. For instance, Hoque et al. (2023) leverage a GRU-based RNN combined with an N-gram language model (unigram, bigram, trigram, 4-gram, and 5-gram) on a Bangla dataset. The researchers focus on enhancing the accuracy and efficiency of language processing for Bangla, tackling challenges such as data sparsity and the complexity of the language. Their proposed model achieves significantly higher accuracy compared to previous approaches (81.22% - 99.78%).

Additionally, the hybrid approach for document-level sentiment analysis is also proposed utilizing the CNN-BiLSTM model, achieving 90.66% accuracy (Rhanoui et al., 2019). They combine Convolutional and Bi-LSTM networks with Doc2Vec embeddings using a corpus dataset of 2003 French news articles (positive, neutral, negative). A hybrid model combining Trie, CNN, and LSTM for Bangla's next sequence generation, addressing limitations posed by traditional N-gram models, is studied (Nobel et al., 2023). As a result, the model captures long-range dependencies and contextual patterns in the Bangla corpus, achieving promising results on a diverse dataset. The effectiveness of different word embedding techniques for next sequence prediction was investigated using a Bengali dataset (Islam et al., 2024). The authors compare LSTM models trained on word2vec (skip-gram and CBOW) and fast Text (skip-gram and CBOW) embeddings against the n-gram models. Their results illustrate the superiority of LSTM-based models, with word2vec skip-gram achieving the highest accuracy (79.72% for N=1) while fast Text models performed slightly lower. The study highlights the importance of contextual understanding for accurate next-word generation in Bengali.

Furthermore, recent studies have increasingly utilized context-aware mechanisms, such as attention mechanisms, to address the shortcomings of traditional machine-learning models. Attention mechanisms enhance the focus on essential features and handle complex data interactions. In the context of morphologically rich languages like Afaan Oromo, attention mechanisms are crucial as they allow the model to focus on key morphological features (such as prefixes or suffixes) that are central to understanding the word structure and meaning. For instance, Peng et al. (2021) introduced a Social Relational Attention LSTM (SRA-LSTM) model to capture social relationships between pedestrians in trajectory prediction. Later on, Tang et al. (2020) proposed an attention-based LSTM combined with genetic algorithms to analyze urban road traffic flow. The multi-head attention mechanisms to enhance multimodal future trajectory prediction are also examined (Kim et al., 2020). These attention mechanisms were also suggested as the optimal solution in textual data analysis (Ayetiran, 2022; Kumar et al., 2023; Le, 2020; Mao et al., 2022; Putelli et al., 2021). For example, Le (2020) introduced methods for sentiment analysis by utilizing a sentiment lexicon and a word embedding technique (Word2vec) with an attention mechanism. In this context, the attention mechanism aids the model in concentrating on important words based on their sentiment score, resulting in improved performance compared to other popular machine learning techniques.

Building on these advancements, our study proposes a synergistic approach that merges the Bidirectional LSTM model with attention mechanisms for Afaan Oromo's next word generation, inspired by similar principles used in trajectory prediction and sentiment analysis. This attention-driven Bi-LSTM model effectively captures temporal sequences by processing information in both forward and backward directions (Bidirectional) while integrating attention mechanisms to emphasize significant features and enhance prediction accuracy. This approach aims to tackle the challenges of accurately predicting the next word in Afaan Oromo by capturing complex language patterns and contextual dependencies. With this approach, we enhance the accuracy of next-word generation and provide insights into the impact of various contextual factors on language generation, aligning with recent advancements in predictive modeling across different domains.

3. Methodology

3.1. Materials

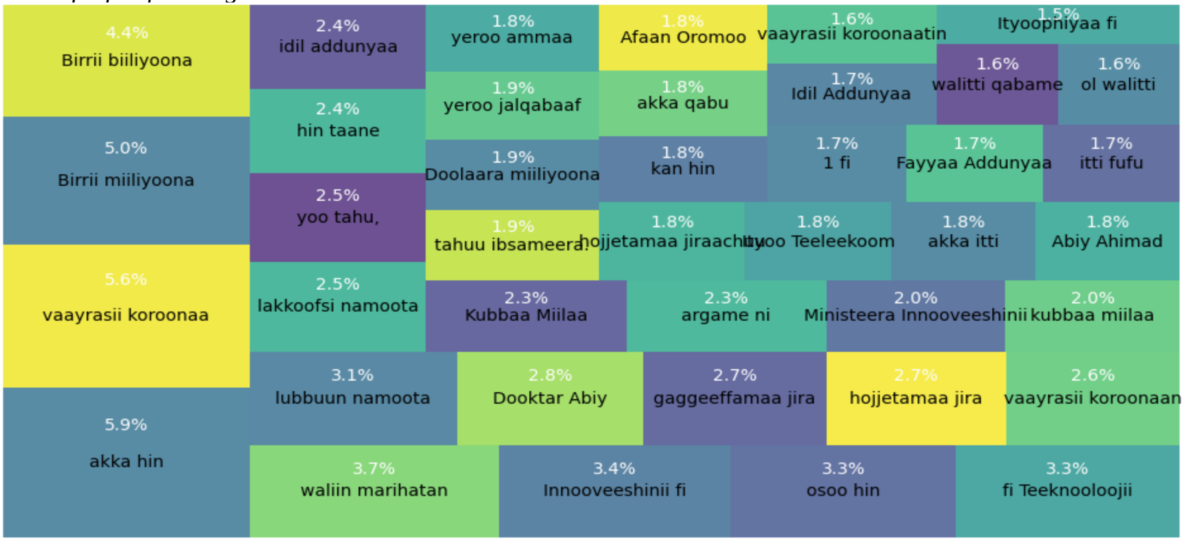
We have compiled a dataset of Afaan Oromo sentences from various sources such as broadcasting Media (BBC, FBC, OMN, OBN) of Afaan Oromo, Social Media such as Facebook and Twitter, Academic books, and Afaan Oromo bible texts. The final dataset is 2 MB of textual data and comprises 510 pages, totaling 201,538 words, of which 20% are used to evaluate model performance. The sentence lengths within our dataset vary, with each sentence containing at least three words and no upper limit on the number of words. Table 1 provides a summary of our data sources.

Table 1
Sources of Our Dataset

Source of Words	Total Words	Total Sentences
Broadcasting media (BBC, FBC, OMN, OBN) of Afaan Oromo.	90,340	9,500
Social Media (Facebook, Twitter)	32,055	2,400
Afaan Oromo Academic Books	49,253	4,000
Afaan Oromo bible books	29,890	2,000

While these sources may introduce certain biases we addressed this during data preprocessing. All data was carefully structured and standardized. In cases where informal or inconsistent language was detected, we corrected sentences to ensure proper grammar and full sentence structures, thereby enhancing the uniformity and quality of the dataset for training. Furthermore, Figure 2 indicates a visual representation of the most common word pairs, enhancing our understanding of linguistic patterns relevant to the next word generation.

Figure 2
Treemap of Top 20 Bigrams in Our Dataset

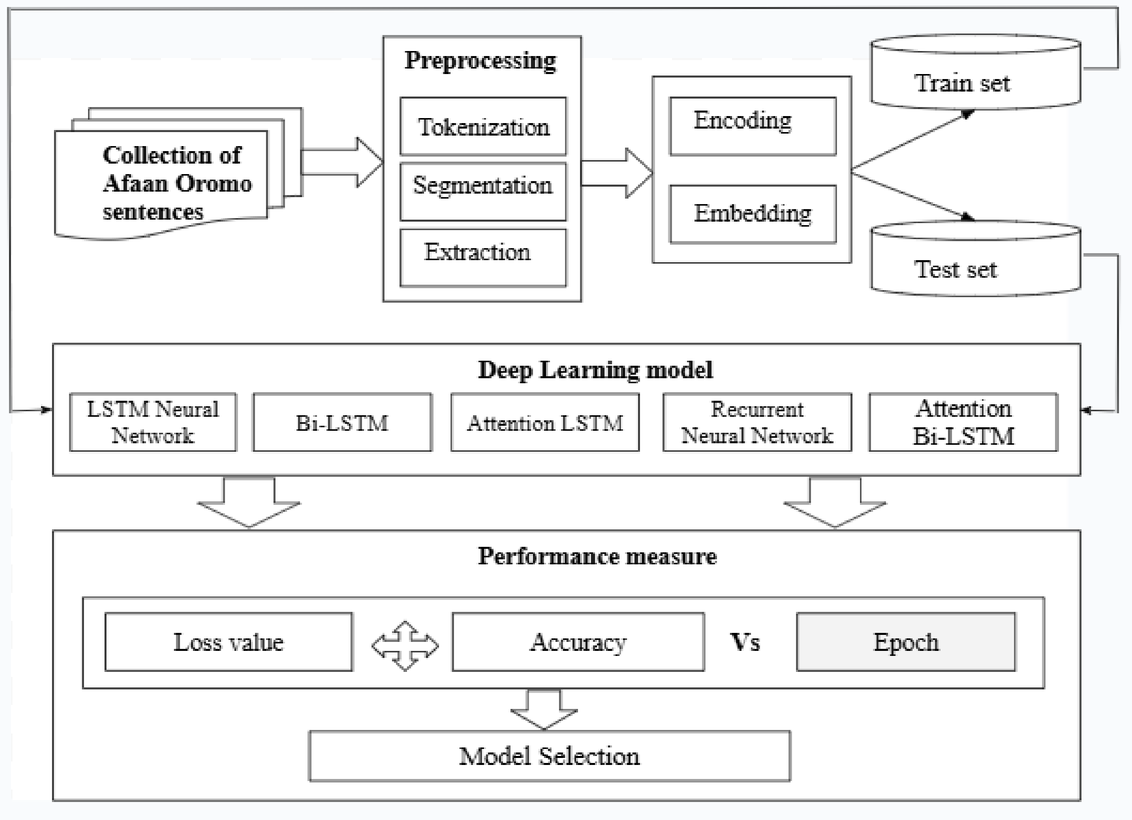


3.2. Procedures

The procedure for developing the Attention-Driven Bi-LSTM model followed several critical steps. After the dataset was collected, we employed a series of Natural Language Processing (NLP) techniques to prepare the data for model training. This included tokenization, where text data was split into individual words or sub-words, and padding, which ensured that all sequences in the dataset had the same length for consistency in model input. For the next-word generation task, we created sequences of tokens, where the predictors were the sequence of words up to the target word. This approach provided the necessary format for training the model to predict the subsequent word given the context of previous words. The details of these preprocessing steps are outlined in Figure 3.

Figure 3

Architecture of Proposed Next-word Generation Model



As shown in Figure 3, several models, including LSTM, Bi-LSTM, Attention-based LSTM, and Attention-based Bi-LSTM, were tested to determine the most effective model for our next-word generation task. Furthermore, to ensure diversity in our datasets, we incorporated data from varied dialects, demonstrating our work's potential impact on the active use and preservation of Afaan Oromo, thereby enhancing digital communication efficiency and preserving the language in digital formats.

3.2.1. LSTM Network

LSTM is a special type of Recurrent Neural Network (RNN) designed to improve the issue of long-term dependencies in sequential data processing (Saha & Senapati, 2020). Unlike standard RNNs that struggle with retaining information for extended periods, LSTMs are explicitly built to overcome this limitation. LSTMs achieve this by incorporating a "cell state" that acts as a conveyor belt, allowing relevant information to flow throughout the network without significant degradation. The network uses various gates (forget gate, input gate, output gate) to control the flow of information within the cell state, ensuring only pertinent data remains (Saha & Senapati, 2020). LSTM maintains a cell state C_t in the time interval t to consistently learn sequential relationships. At each time step, LSTM considers C_{t-1}, h_{t-1}, and x_t as input, and the input gate determines whether the preceding information (h_{t-1} and x_t)

is passed to the cell state. If the forget gate f_t is activated, the network will discard the previous memory cell C_{t-1}. The output gate O_t controls the output of the memory cell. The whole process of the LSTM unit is formulated as follows:

$$f_t = \sigma\,(W_f * [h_{t-1}, x_t] + b_f \qquad (1)$$

$$i_t = \sigma\,(W_i * [h_{t-1}, x_t] + b_i \qquad (2)$$

$$\hat{C}_t = tanh\,(W_C * [h_{t-1}, x_t] + b_C \qquad (3)$$

$$o_t = \sigma\,(W_o * [h_{t-1}, x_t] + b_o \qquad (4)$$

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

$$h_t = o_t * tanh(C_t) \qquad (6)$$

Where h_t denotes the hidden state of the LSTM unit at the time step t, $*$ represents the element-wise multiplication operation, *tanh* is the hyperbolic tan function, and σ is the sigmoid activation function. W_v, b_v $(v \in \{f, i, C, o\})$ are parameters to be learned.

3.2.2. Bi-LSTM Network

Bi-LSTM, or Bidirectional Long Short-Term Memory, is an extension of the LSTM architecture designed to improve the modeling of sequential data by processing information in both forward and backward directions, as in Figure 4. This bidirectional strategy enables the network to capture dependencies from both preceding and subsequent contexts, which is especially useful for comprehending the full context of a sequence. Hence, Bi-LSTMs are specifically designed to address the long-term dependency issue by leveraging two separate LSTM networks: one for processing the sequence in the forward direction and one for the reverse direction (Li et al., 2020).

Forward LSTM: processes the input sequence from left to right (preceding contexts).

$$f_t = \sigma\,(W_f * [\overrightarrow{h_{t-1}}, x_t] + b_f \qquad (7)$$

$$i_t = \sigma\,(W_i * [\overrightarrow{h_{t-1}}, x_t] + b_i \qquad (8)$$

$$\hat{C}_t = tanh\,(W_C * [\overrightarrow{h_{t-1}}, x_t] + b_C \qquad (9)$$

$$o_t = \sigma\,(W_o * [\overrightarrow{h_{t-1}}, x_t] + b_o \qquad (10)$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \qquad (11)$$

$$\overrightarrow{h_t} = o_t * tanh(C_t) \qquad (12)$$

Backward LSTM: processes the input sequence from right to left (subsequent contexts).

$$\acute{f}_t = \sigma\,(\acute{W}_f * [\overleftarrow{h_{t+1}}, x_t] + \acute{b}_f \qquad (13)$$

$$\acute{i}_t = \sigma\,(\acute{W}_i * [\overleftarrow{h_{t+1}}, x_t] + \acute{b}_i \qquad (14)$$

$$\acute{\hat{C}}_t = tanh\,(\acute{W}_C * [\overleftarrow{h_{t+1}}, x_t] + \acute{b}_C \qquad (15)$$

$$\acute{o}_t = \sigma\,(\acute{W}_o * [\overleftarrow{h_{t+1}}, x_t] + \acute{b}_o \qquad (16)$$

$$\acute{C}_t = \acute{f}_t * C_{t+1} + i_t * \acute{\hat{C}}_t \qquad (17)$$

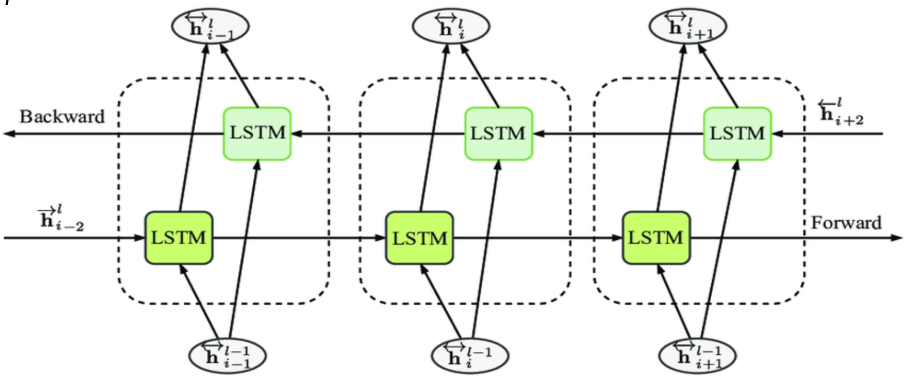
$$\overleftarrow{h_t} = \acute{o}_t * tanh(\acute{C}_t) \qquad (18)$$

Where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ denote the hidden states of the forward and backward LSTM units at time step *t*, respectively. While * denotes the element-wise multiplication operation, *tanh* is the hyperbolic tangent function, and σ is the sigmoid activation function. *W* and *b* parameters are learned during training.

Final Hidden State: The outputs of these two LSTMs are then combined to provide a comprehensive representation of the input sequence. That is, the final hidden state(h_t) for a given time step *t* is the concatenation of the forward and backward hidden states:

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \tag{19}$$

Figure 4
Architecture of Bi-LSTM Network



3.2.3. Attention-Based LSTM Network

Attention-Based LSTM Network is an enhancement over Long Short-Term Memory (LSTM) integrating an attention mechanism to address the limitations of standard LSTMs (Li et al., 2020). Although LSTM networks excel at capturing sequential dependencies, they consider each element in the sequence equally. However, in many natural language processing tasks, including next-word generation, not all parts of the input sequence are equally significant for predicting the output. For instance, in the sentence "*The quick brown fox jumps over the lazy dog*," the words "*quick*", "*brown*", and "*fox*" are more relevant for predicting the word "*jumps*" than the words "*the*" and "lazy." So, the attention mechanisms handle this limitation by assigning weights to different parts of the input sequence, allowing the model to focus on the most important information (Ayetiran, 2022; Kumar et al., 2023; Le, 2020; Mao et al., 2022; Putelli et al., 2021). These weights are then used to compute a context vector, which is a weighted sum of the hidden states generated by the LSTM. Below are the steps involved in the attention mechanism. It uses the hidden state generated from the LSTM's output as an input (Li et al., 2020). That is,

The hidden state at time t, which is generated as LSTM layer output, is:

$$h_t = LSTM(x_t, h_{t-1}) \tag{20}$$

Calculate the attention weight using the softmax function:

$$alpha_t = softmax(v^T{}_a * \tanh(W_a * h_t + b_a)) \tag{21}$$

Compute the context vector (c) as a weighted sum of the hidden states:

$$c = \sum (alpha_t * h_t) \tag{22}$$

Calculate the predicted probability distribution over the vocabulary using softmax.

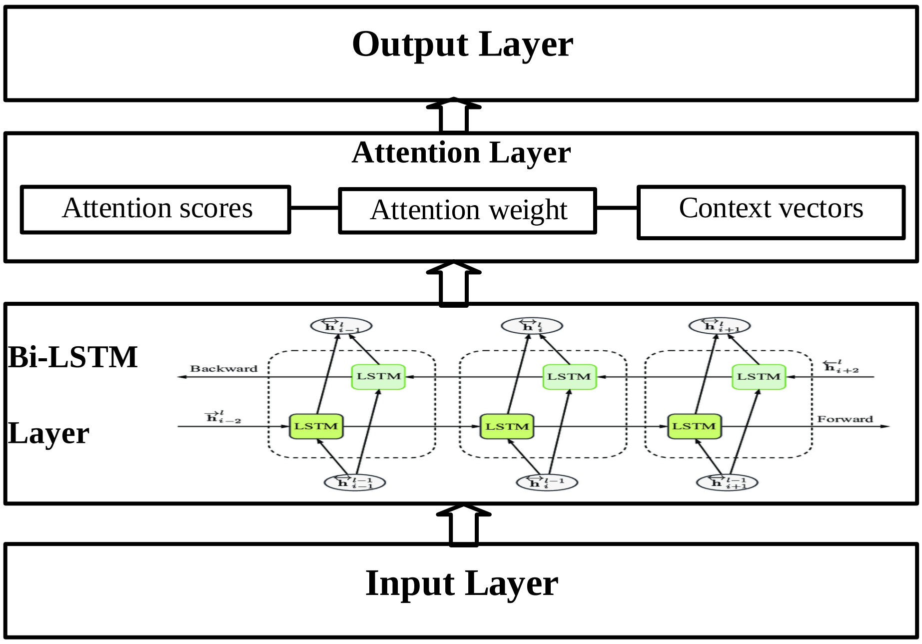
$$y_{hat} = softmax(W_s * [h_n, c] + b_s) \tag{23}$$

Where h_t is the hidden state at time step t, x_t is the input word embedding at time step t, h_n is the last hidden state of the LSTM, y_{hat} is the predicted probability distribution over the vocabulary (W_s, b_s, v_a, W_a, *and* b_a) are learnable parameters.

3.2.4. Attention-Based Bi-LSTM Network

The Attention-Based Bi-LSTM Network enhances the traditional Bi-LSTM model by incorporating an attention mechanism to address the limitations of standard Bi-LSTMs (Chopannejad et al., 2024). Although Bi-LSTMs are effective at capturing sequential dependencies by processing sequences in both forward and backward directions, they treat all parts of the sequence with equal importance. In various natural language processing tasks, such as next-word generation, different parts of the input sequence contribute differently to the prediction. Attention-based mechanisms improve this by assigning varying weights to different parts of the input sequence, allowing the model to focus on the most relevant information. These weights are used to compute a context vector that combines information from both forward and backward hidden states generated by the Bi-LSTM.

Figure 5
Architecture of Attention-Driven Bi-LSTM Used in Our Proposed Approach



Below are the steps involved in the attention mechanism within a Bi-LSTM framework:

1. **Bi-LSTM Hidden States:** The hidden states from the forward LSTM and backward LSTM at time step **t** are:

$$\overrightarrow{h_t} = Forward\_LSTM(x_t, \overrightarrow{h_{t-1}}) \quad (24)$$

$$\overleftarrow{h_t} = Backward\_LSTM(x_t, \overleftarrow{h_{t+1}}) \quad (25)$$

2. **Concatenate Hidden States:** The combined hidden state at time step **t** is:

$$h_t = [\overrightarrow{h_t} ; \overleftarrow{h_{t+1}}] \quad (26)$$

3. **Calculate Attention Weights:** Compute the attention weights for each time step using the context vector approach. Here, v_a, W_a, and b_a are learnable parameters:

$$a_t = softmax({v^T}_a * \tanh(W_a * h_t + b_a)) \quad (27)$$

4. **Compute Context Vector:** The context vector **c** is a weighted sum of the hidden states, taking into account the attention weights:

$$c = \sum_t (a_t * h_t) \tag{28}$$

5. **Predict Probability Distribution:** Finally, calculate the predicted probability distribution over the vocabulary using the context vector **c** and the concatenated hidden state:

$$\hat{y} = softmax(W_s * [h_n ; c] + b_s) \tag{29}$$

Where W_s and b_s are learnable parameters.

Algorithm: Attention-Driven Bi-LSTM Model for Afaan Oromo Next Sequence Generation

---

Start

1. ***Input****: Text data sequence*
2. ***Preprocess Data****:*
   - *Tokenization: Convert text to token indices.*
   - *Padding: Pad sequences to uniform length.*
   - *Create Predictors and Labels:*
     - *Predictors = Padded_sequences[:, :-1]*
     - *Labels = to_categorical(Padded_sequences[:, -1], num_classes=Totalwords)*
3. ***Build Model****:*
   - *Input Layer: Input(shape=(L-1,))*
   - *Embedding Layer: Embedding(Totalwords, 100)*
   - *Bidirectional LSTM Layer: Bidirectional(LSTM(100, return_sequences=True))*
   - *Attention Mechanism:*
     - *Query = LSTM(100, return_sequences=True),*
     - *Key = LSTM(100, return_sequences=True),*
     - *Value = LSTM(100, return_sequences=True)*
     - *Attention Computation: Attention_output = Attention ([Query, Value])*
   - *Concatenation: Concatenate([Attention_output, LSTM_output])*
   - *Final LSTM Layer: LSTM(100)*
   - *Dense Output Layer: Dense(Totalwords, activation='softmax')*
4. ***Compile Model****:*
   - *Loss Function: Categorical crossentropy*
   - *Optimizer: Adam*
5. ***Train Model****:*
   - *Fit Model: model.fit(Predictors, Labels, epochs=#)*

6. ***Save Model***:

Save: model.save('model_ABiLSTM.keras')

End

3.3. Activation Functions Used in the Proposed Model

In Attention-Based Bi-LSTM networks designed for sequence generation tasks, such as next-word generation, the choice of activation functions is crucial for optimizing model performance, particularly in low-resource languages. In this work, I have utilized the sigmoid function and the softmax functions at different stages of the network, each serving distinct purposes integral to the model's performance.

3.3.1. Sigmoid Activation in LSTM Units

In the Bi-LSTM architecture, the LSTM units utilize gating mechanisms using forget gate, input gate, and output gate to control the flow of information across sequences (Chopannejad et al., 2024). Each of these gates relies on the sigmoid activation function (Lv et al., 2019). The sigmoid function's ability to produce outputs between 0 and 1 allows these gates to effectively regulate the retention, discarding, and passing of information, which is essential for capturing long-range dependencies and maintaining gradient stability during training. This functionality is particularly beneficial in low-resource languages like Afaan Oromo, where limited data may make it challenging to learn complex sequential patterns.

3.3.2. Softmax Activation in the Output Layer

In our next word generation task, we also utilized the softmax function in the final layer of the model to transform the raw network predictions into a normalized probability distribution over the vocabulary (Yamashita et al., 2018). This enables the model to output a probabilistic prediction for each potential next word, facilitating accurate multi-class classification and making the predictions more interpretable. For low-resource languages, where data scarcity can exacerbate difficulties in learning and predicting, these activation functions contribute significantly to the effectiveness of models by enabling them to handle sparse data more robustly and provide reliable predictions despite limited training examples. That is, given a vector of raw scores $[z_1, z_2 . . . z_n]$, the softmax function computes the probability for each class ***i*** as:

$$Softmax(y_i) = \frac{e^{z^i}}{\sum_{j=1}^{k} e^{z_j}} \qquad (30)$$

Where softmax(y_i) is the probability of the i^{th} word being the next word in the sequence), z_i represents the raw score for class *i, and e* is the base of the natural logarithm, used here to exponentiation of the raw scores, ensuring that all probabilities are positive. The denominator,$\frac{e^{z^i}}{\sum_{j=1}^{k} e^{z_j}}$, is the sum of exponentiated raw scores for all possible classes *j*. This normalization step ensures that the sum of all probabilities across the vocabulary equals 1.

4. Results

4.1. Experimental Setup

All experiments detailed in this paper are carried out on a standardized computer setup featuring a 64-bit Ubuntu 22.04.4 LTS Operating System, an Intel® Core™ i7 CPU, and 32GB of memory. The experiments are run within a Python 3.7 environment. In addition to the hardware and software specifications, below is the list of hyperparameters utilized in our model configurations.

Table 2

Hyperparameters Utilized in Our Selected Model Configurations

No.	Hyperparameter	Value
1.	Embedding Dimension	100
2.	Bidirectional LSTM Units	100 (for each LSTM layer)
3.	Number of Epochs	30
4.	Batch Size	64
5.	Loss Function	Categorical Cross entropy
6.	Optimizer | Learning Rate	Adam Optimizer | 0.001
7.	Dropout Rate	0.3
8.	Total Unique Words	112,000 words

4.2. Model Performance Evaluation

In this study, we evaluated five neural network architectures, LSTM, Attention-LSTM, Bi-LSTM, Attention-Based Bi-LSTM, and RNN, on their ability to predict the next word in sequences of Afaan Oromo text. The performance metrics, including accuracy and loss, are summarized in Table 2 and discussed below. Among the five models, the Attention-based Bi-LSTM achieved the highest accuracy of 95.0% and the lowest loss of 0.27, demonstrating its superior ability to predict the next word with high precision and minimal error. The LSTM model followed with an accuracy of 94.0% and a loss of 0.38, reflecting strong performance with a reliable balance between prediction accuracy and error minimization.

Table 3

Summary of the Performance Results of Each Model

Metrics	LSTM	Attention - LSTM	Bi-LSTM	Attention Bi-LSTM	RNN
Accuracy	0.94	0.88	0.90	**0.95**	0.90
Loss	0.38	0.91	0.84	**0.27**	0.53

The Bi-LSTM model recorded an accuracy of 90.0% and a loss of 0.84, showing improved context understanding over simpler models but falling short of the top-performing models. The RNN model also achieved an accuracy of 90.0% but with a higher loss of 0.53, indicating moderate performance with less effectiveness in error reduction compared to more advanced architectures. The Attention-based LSTM model, despite using attention mechanisms, had the lowest accuracy at 88.0% and the highest loss at 0.91, suggesting that its attention mechanisms did not significantly enhance performance in this context without further improvements.

4.3. Impacts of the Model Type and Training Epochs

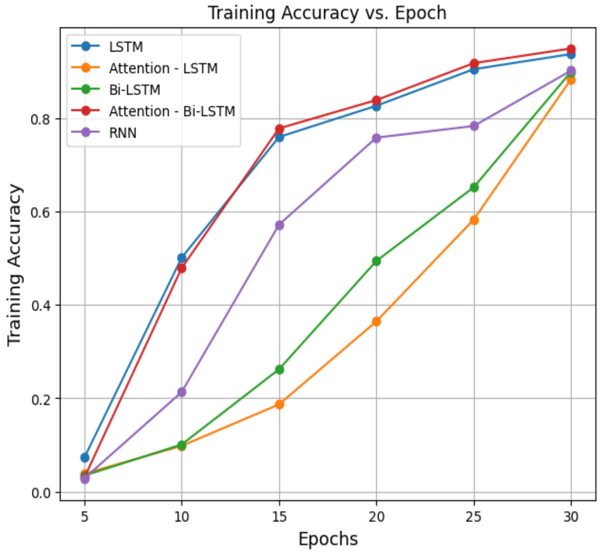
To evaluate model types and training epochs' impact on the performance, we have compared five different neural network architectures: LSTM, Attention-LSTM, Bi-LSTM, Attention-Based Bi-LSTM, and RNN. We analyzed their training accuracy and loss, precision, recall, and F1-score over a range of epochs (5, 10, 5, 20, 25, and 30).

4.3.1. Training Accuracy

The training accuracy, shown in Figure 6, demonstrates how well each model learns over increasing epochs. The LSTM model starts with a low accuracy but improves significantly, reaching an accuracy of 94.2% by epoch 30. The Attention-LSTM model also shows substantial improvement, achieving an accuracy of 88.4% by the final epoch, though it lags behind LSTM. The Bi-LSTM model shows an initial accuracy similar to the LSTM but eventually reaches a plateau at 90.0% accuracy. The Attention-Based Bi-LSTM model, however, exhibits the highest accuracy, peaking at 95.0% by epoch 30. The RNN model, despite showing improvements, achieves an accuracy of 90.2%, which is lower than that of the Attention-Based Bi-LSTM.

Figure 6

Performance of Each Model in Terms of Accuracy vs Epoch

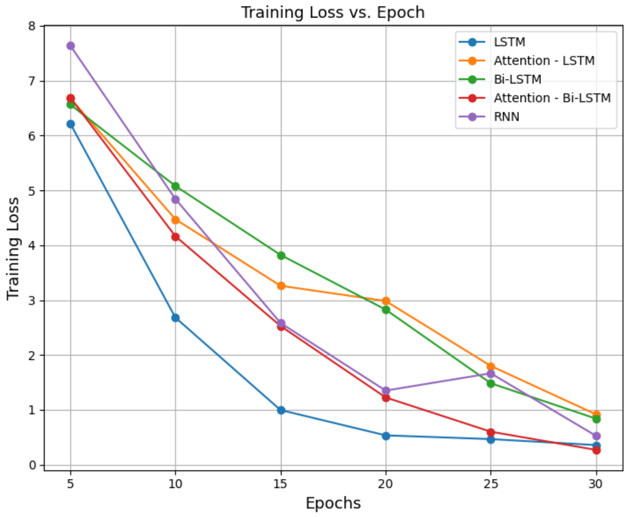


4.3.2. Training Loss

The training loss curves, illustrated in Figure 7, reveal the convergence characteristics of each model. The LSTM model shows a steady decline in loss, reaching 0.3584 by epoch 30, indicating effective learning. In comparison, the Attention-LSTM model experiences a slower reduction in loss, ending at 0.9181. The Bi-LSTM model exhibits a more rapid decline in loss compared to LSTM, reaching 0.8402. The Attention-Based Bi-LSTM achieves the lowest final loss of 0.2709, reflecting its superior performance in minimizing error during training. Conversely, the RNN model shows a significant reduction in loss, ending at 0.5275, but still higher than the other models.

Figure 7

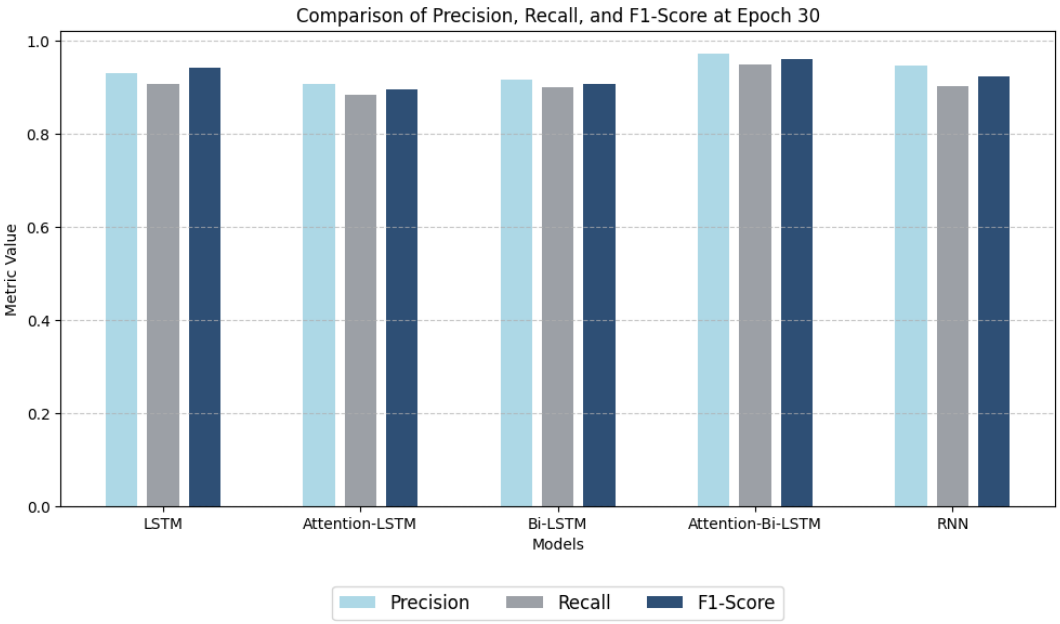
Performance of Each Model in Terms of Loss vs Epoch



In addition to accuracy and loss metrics, the performance of each model is assessed using precision, recall, and F1-score to evaluate the predictive power in the Afaan Oromo language, as shown in Figure 8. The effectiveness of the attention-driven Bi-LSTM network in predicting the next word stems from its unique ability to employ attention mechanisms, which dynamically assess the importance of different parts of the input sequence during prediction. The results demonstrate that the Attention-Bi-LSTM model achieved the highest performance in precision (0.9729), recall (0.9500), and F1-score (0.9613), significantly outperforming the other models. This indicates that the attention mechanism plays a crucial role in improving prediction accuracy by enabling the model to focus on relevant context, addressing the limitations of traditional models that treat all input sequences equally without accounting for varying importance.

Figure 8

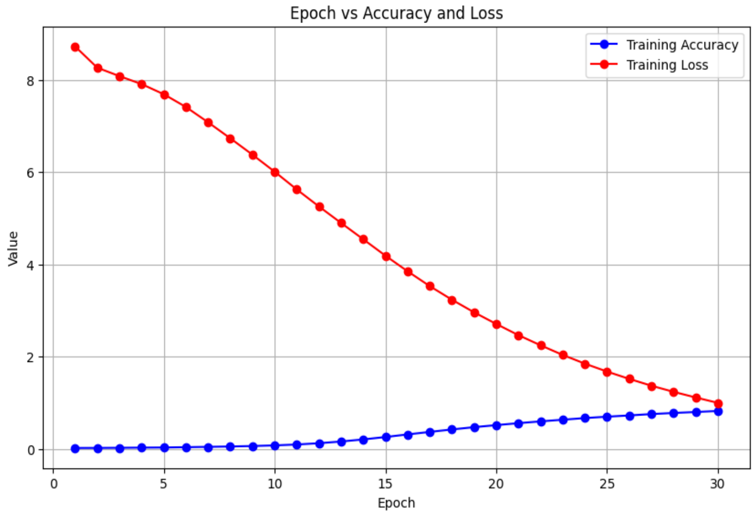
Performance Comparison of Precision, Recall and F1-Score



Moreover, the attention-driven Bi-LSTM effectively captures long-range dependencies and enhances the handling of linguistic features in Afaan Oromo. Its performance metrics indicate strong performance in managing complex word formations and structural variations, as reflected in high F1 scores. Overall, the Attention-Based Bi-LSTM emerges as the most effective approach for next-word generation, emphasizing the importance of advanced neural network architectures for languages with complex structures. Given the significance of Afaan Oromo as the most widely spoken Cushitic language, improving language processing tools is crucial. Additionally, Figure 9 provides a comprehensive overview of the model's learning process, showing that the accuracy of the Attention-Driven Bi-LSTM improves with increasing epochs while training loss decreases over the same period.

Figure 9

Attention-driven Bi-LSTM Performance



5. Discussion

In this study, we tested the performance of five distinct neural network architectures: LSTM, Attention-LSTM, Bi-LSTM, Attention-Based Bi-LSTM, and RNN for next-word generation in Afaan Oromo text sequences. Out of these models, the Attention-Based Bi-LSTM had the best performance, achieving the highest accuracy of 95.0% and the lowest loss of 0.27. This performance supports the findings ofAmbulgekar et al. (2021) and indicates the model's ability to handle the linguistic complexities found in Afaan Oromo, such as its rich morphology, diverse word forms, and unique sentence structure. The attention mechanism leveraged in this model enabled it to focus on key contextual features, while the bidirectional feature facilitated a deeper understanding of the sequential relationships within the text. In contrast, the LSTM model, which also performed strongly, achieved an accuracy of 94.0% and a loss of 0.38. This indicates that while the LSTM offers a reliable balance between prediction accuracy and error minimization, it is slightly less capable of capturing the intricate contextual dependencies like morphology that the Attention-Based Bi-LSTM excels at.

The other models, Bi-LSTM and RNN, both achieved an accuracy of 90.0% and similar loss values, indicating a more modest performance compared to the Attention-Based Bi-LSTM models. However, Sharma et al. (2019) found Bi-LSTM achieved an accuracy of 81.07%, showcasing our results signify a considerable advancement. These results reflect the trade-offs between model complexity and performance, with Bi-LSTM offering an improvement over the simpler LSTM model but still lagging behind the more advanced attention-driven architectures. The RNN model also demonstrated a similar performance but struggled to keep up with the advanced capabilities of models that incorporate attention mechanisms. The Attention-based LSTM model, despite its use of an attention mechanism, showed the weakest accuracy at 88.0% and the highest loss at 0.91. This indicates that while attention mechanisms have the potential to improve performance, their integration in this model was not fully optimized, showing the need for further refinement to better improve the power of attention mechanisms in future iterations.

Moreover, the findings of this study underscore the significant impact that advanced neural network architectures, particularly Attention-Based Bi-LSTM models, can have on next-word generation and prediction, especially for languages with complex structures like Afaan Oromo. This impact aligns with findings from Wangchuk et al. (2023) and Sumathy et al. (2023), highlighting the effectiveness of advanced models in managing the linguistic intricacies of Afaan Oromo and the challenges that simpler frameworks struggle to address. By leveraging attention mechanisms, the Attention-Based Bi-LSTM excels in capturing long-range dependencies, including morphological structures, syntactic patterns, and contextual information that is crucial for accurate predictions. The results of this study indicate that these advanced architectures offer substantial improvements over simpler models, establishing them as highly effective tools for natural language processing tasks in languages with similarly complex grammatical features. This performance highlights the potential of attention-driven models to make significant contributions to enhancing the quality and accuracy of text prediction systems in such languages.

In future work, there are several promising approaches to further improve model performance and, hence, increase the generalizability of the findings. One key approach is leveraging attention mechanisms within the models by exploring more sophisticated hybrid models that combine the strengths of different architectures. Such hybrid models could further improve prediction accuracy by utilizing additional layers for contextual understanding and by integrating more diverse data representations, as seen in (Hoque et al., 2023). Additionally, validating these findings on larger and more diverse datasets will not only improve the robustness of the model but also ensure its effectiveness across different domains and real-world applications. Although the dataset of 201,538 words used in this study is relatively small for training deep models, this limitation may lead to challenges like overfitting and poor generalization. Future work could address this by employing data augmentation techniques such as paraphrasing or noise injection. Transfer learning could also be considered, using pre-trained models from other languages or multilingual models to mitigate the impact of the small dataset. However, transfer learning was not applied in this study due to the lack of pre-trained models for Afaan Oromo. Addressing semantic ambiguities in Afaan Oromo, such as subtle variations in word forms (e.g., "Gadaa" versus "Gaddaa"), is another key area that needs further investigation. Tackling these challenges could lead to even greater accuracy and clarity in next-word generation systems, particularly in languages where small changes in wording can profoundly impact or alter the meaning.

Disclosure Statement

The authors claim no conflict of interest.

Funding

The research did not receive any specific grants from funding agencies.

References

Ambulgekar, S., Malewadikar, S., Garande, R., & Joshi, B. (2021). Next words prediction using recurrent neural networks. *ITM Web of Conferences*, *40*, Article 3034. https://doi.org/10.1051/itmconf/20214003034

Ayetiran, E. F. (2022). Attention-based aspect sentiment classification using enhanced learning through cnn-Bilstm networks. *Knowledge-Based Systems*, *252*, Article 109409. https://doi.org/10.1016/j.knosys.2022.109409

Barman, P. P., & Boruah, A. (2018). A RNN based approach for next word prediction in Assamese phonetic transcription. *Procedia Computer Science*, *143*, 117–123. https://doi.org/10.1016/j.procs.2018.10.359

Bhuvaneswari, A., Jones Thomas, J. T., & Kesavan, P. (2019). Embedded bi-directional GRU and LSTM learning models to predict disasters on Twitter data. *Procedia Computer Science*, *165*, 511–516. https://doi.org/10.1016/j.procs.2020.01.020

Bickel, S., Haider, P., & Scheffer, T. (2005). *Predicting sentences using N-gram language models*. CiteSeer X (the Pennsylvania State University). https://doi.org/10.3115/1220575.1220600

Chopannejad, S., Roshanpoor, A., & Sadoughi, F. (2024). Attention-assisted hybrid CNN-BILSTM-BiGRU model with SMOTE–Tomek method to detect cardiac arrhythmia based on 12-lead electrocardiogram signals. *Digital Health*, *10*, 1-20. https://doi.org/10.1177/20552076241234624

Ganai, A. F., & Khursheed, F. (2019). Predicting next word using RNN and LSTM cells: Statistical language modeling. In S. P. Ghrera & P. K. Gupta (Eds.), *Fifth International Conference on Image Information Processing* (pp. 469-474). IEEE Xplore. https://doi.org/10.1109/ICIIP47207.2019.8985885

Garoma, E. T. (2024). Demonstratives in Afaan Oromoo. *Cogent Arts and Humanities*, *11*(1), Article 2297494. https://doi.org/10.1080/23311983.2023.2297494

Gemeda, G. (2023, September 15). *The history and politics of the Qubee alphabet*. Curate Oromia. https://curateoromia.com/the-history-and-politics-of-the-qubee-alphabet/

Hoque, A., Jahan, B., Paul, S. C., Zabu, Z. A., Mondal, R., & Akter, P. (2023). Next words prediction and sentence completion in Bangla language using GRU-Based RNN on N-Gram language model. *Journal of Data Analysis and Information Processing*, *11*(04), 388–399. https://doi.org/10.4236/jdaip.2023.114020

Iqbal, T., & Qureshi, S. (2020). The survey: Text generation models in deep learning. *Journal of King Saud University - Computer and Information Sciences*, *34*(6), 2515–2528. https://doi.org/10.1016/j.jksuci.2020.04.001

Islam, M. R., Amin, A., & Zereen, A. N. (2024). *Enhancing Bangla language next word prediction and sentence completion through extended RNN with Bi-LSTM model on N-gram language*. ArXiv. https://arxiv.org/abs/2405.01873

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2022). Natural language processing: State of the art, current trends and challenges. *Multimedia Tools and Applications*, *82*(3), 3713–3744. https://doi.org/10.1007/s11042-022-13428-4

Kim, H., Kim, D., Kim, G., Cho, J., & Huh, K. (2020). Multi-head attention based probabilistic vehicle trajectory prediction. In B. Morris, F. Garcia, M. Trivedi, & K. Driggs-Campbell (Eds.), *IEEE Intelligent Vehicles Symposium* (IV, pp. 1720-1725). IEEE. https://doi.org/10.1109/iv47402.2020.9304741

Kumar M., P., Babu, R. L., Sunil, G., Sampathkumar, J., & Chakraborty, S. (2023). Attention-based deep learning algorithm in natural language processing for optical character recognition. In J. R. & S. G. (Eds.), *International Conference on Evolutionary Algorithms and Soft Computing Techniques* (pp. 1-5). IEEE. https://doi.org/10.1109/easct59475.2023.10393635

Le, T. (2020). An attention-based deep learning method for text sentiment analysis. In H. R. Arabnia, L. Deligiannidis, F. G. Tinetti, & Q. Tran (Eds.), *International Conference on Computational Science and Computational Intelligence* (pp. 282-286). IEEE. https://doi.org/10.1109/csci51800.2020.00054

Li, W., Qi, F., Tang, M., & Yu, Z. (2020). Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification. *Neurocomputing*, *387*, 63–77. https://doi.org/10.1016/j.neucom.2020.01.006

Liu, Z., & Sun, M. (2023). Representation learning and NLP. In Z. Liu, Y. Lin, & M. Sun (Eds.), *Representation learning for natural language processing* (pp. 1–27). Springer. https://doi.org/10.1007/978-981-99-1600-9_1

Lv, Q., Chen, H., Zhong, W., Wang, Y., Song, J., Guo, S., Qi, L., & Chen, C. Y. (2019). A multi-task group Bi-LSTM networks application on electrocardiogram classification. *IEEE Journal of Translational Engineering in Health and Medicine*, *8*, 1–11. https://doi.org/10.1109/jtehm.2019.2952610

Mao, Y., Zhang, Y., Jiao, L., & Zhang, H. (2022). Document-level sentiment analysis using attention-based bi-directional long short-term memory network and two-dimensional convolutional neural network. *Electronics*, *11*(12), Article 1906. https://doi.org/10.3390/electronics11121906

Nobel, S. N., Sultana, S., Tasir, M. A. M., & Rahman, M. S. (2023). Next word prediction in Bangla using hybrid approach. In C. Shahnaz, S. A. Fattah, & M. Hoque (Eds.), *26th International Conference on Computer and Information Technology* (pp. 1-6). IEEE. https://doi.org/10.1109/iccit60459.2023.10441580

Patil, M., Dere, M., Visave, M., Sumit Patil, M., & Rumale, A. (2024.). Next word prediction using recurrent neural network. *Industrial Engineering Journal*, *53*(5), 291-301.

Peng, Y., Zhang, G., Shi, J., Xu, B., & Zheng, L. (2021). *SRA-LSTM: Social relationship attention LSTM for human trajectory prediction*. ArXiv. https://arxiv.org/abs/2103.17045

Putelli, L., Gerevini, A. E., Lavelli, A., Maroldi, R., & Serina, I. (2021). Attention-based explanation in a deep learning model for classifying radiology reports. In A. Tucker, P. Henriques Abreu, J. Cardoso, P. Pereira Rodrigues, & D. Riaño (Eds.), *Lecture notes in computer science* (Vol. 12721, pp. 367–372). Springer. https://doi.org/10.1007/978-3-030-77211-6_42

Rathee, V., & Yede, S. (2023). A machine learning approach to predict the next word in a statement. In B. Paulchamy (Ed.), *4th International Conference on Electronics and Sustainable Communication Systems* (pp. 1604-1607). IEEE. https://doi.org/10.1109/icesc57686.2023.10193001

Rhanoui, M., Mikram, M., Yousfi, S., & Barzali, S. (2019). A CNN-BiLSTM model for document-level sentiment analysis. *Machine Learning and Knowledge Extraction*, *1*(3), 832–847. https://doi.org/10.3390/make1030048

Saha, B. N., & Senapati, A. (2020). Long short term memory (LSTM) based deep learning for sentiment analysis of English and Spanish data. In G. Saha, S. Paul, & J. K. Verma (Eds.), *International Conference on Computational Performance Evaluation* (pp. 442-446). IEEE. https://doi.org/10.1109/compe49325.2020.9200054

Sharma, R., Goel, N., Aggarwal, N., Kaur, P., & Prakash, C. (2019). Next word prediction in Hindi using deep learning techniques. In J. Mathew & W. Ku (Eds.), *International Conference on Data Science and Engineering* (pp. 55-60). IEEE. https://doi.org/10.1109/ICDSE47409.2019.8971796

Sumathy, R., Sohail, S. F., Ashraf, S., Reddy, S. Y., Fayaz, S., & Kumar, M. (2023). Next word prediction while typing using LSTM. In L. P. Thangavelu (Ed.), *8th International Conference on Communication and Electronics Systems* (pp. 167-172). IEEE. https://doi.org/10.1109/icces57224.2023.10192602

Tang, J., Zeng, J., Wang, Y., Yuan, H., Liu, F., & Huang, H. (2020). Traffic flow prediction on urban road network based on license plate recognition data: Combining attention-LSTM with genetic algorithm. *Transportmetrica a Transport Science*, *17*(4), 1217–1243. https://doi.org/10.1080/23249935.2020.1845250

Tesema, W., & Tamirat, D. (2017). Investigating Afan Oromo language structure and developing effective file editing tool as plug-in into Ms word to support text entry and input methods. *American Journal of Computer Science and Engineering Survey*, *5*(1). https://www.primescholars.com/articles/investigating-afan-oromo-language-structure-and-developing-effective-file-editing-tool-as-plugin-into-ms-word-to-support-92623.html

Walga, T. K. (2021). Prospects and challenges of Afan Oromo: A commentary. *Theory and Practice in Language Studies*, *11*(6), 606–612. https://doi.org/10.17507/tpls.1106.03

Wangchuk, K., Wangchuk, T., & Namgyel, T. (2023). Dzongkha next words prediction using bidirectional LSTM. *Bhutan Journal of Research and Development*, *2*, 1-17. https://doi.org/10.17102/bjrd.rub.se2.038

Yamashita, R., Nishio, M., Gian, R. K. D. & Togashi, K. (2018). Convolutional neural networks: An overview and application in radiology. *Insights Imaging*, *9*(4), 611–629. https://doi.org/10.1007/s13244-018-0639-9