Complex & Intelligent Systems (2025) 11:10
https://doi.org/10.1007/s40747-024-01631-9

ORIGINAL ARTICLE

Empowering Urdu sentiment analysis: an attention-based stacked CNN-Bi-LSTM DNN with multilingual BERT

Lal Khan[1] · Atika Qazi[2] · Hsien-Tsung Chang[3,4,5] · Mousa Alhajlah[6] · Awais Mahmood[6]

Received: 11 December 2023 / Accepted: 1 September 2024 / Published online: 9 November 2024

Abstract

Sentiment analysis (SA) as a research field has gained popularity among the researcher throughout the globe over the past 10 years. Deep neural networks (DNN) and word vector models are employed nowadays and perform well in sentiment analysis. Among the different deep neural networks utilized for SA globally, Bi-directional long short-term memory (Bi-LSTM), BERT, and CNN models have received much attention. Even though these models can process a wide range of text types, Because DNNs treat different features the same, using these models in the feature learning phase of a DNN model leads to the creation of a feature space with very high dimensionality. We suggest an attention-based, stacked, two-layer CNN-Bi-LSTM DNN to overcome these glitches. After local feature extraction, by applying stacked two-layer Bi-LSTM, our proposed model extracted coming and outgoing sequences by seeing sequential data streams in backward and forward directions. The output of the stacked two-layer Bi-LSTM is supplied to the attention layer to assign various words with varying values. A second Bi-LSTM layer is constructed atop the initial layer in the suggested network to increase performance. Various experiments have been conducted to evaluate the effectiveness of our proposed model on two Urdu sentiment analysis datasets named as UCSA-21 and UCSA, and an accuracies of 83.12% and 78.91% achieved, respectively.

Keywords Sentiment analysis · Word embedding · Machine learning · Deep learning · Bi-LSTM · CNN · Attention · BERT

Introduction and background

The evolving habit of social networking platforms, such as Facebook, YouTube, Twitter, and Instagram, has stimulated and allowed the extensive broadcasting of data and sentiments on dilemmas, retail, strategies, and services. In social networks, individuals can share and express their feelings using various data types, such as tweets, comments, images, videos, and audio clips. A tremendous amount of social networking data is being made on various social media sites daily. This big data imitates the sentiment propensities of individuals towards multiple aspects of life, like social subjects, politics, and business [1].

The English dialect is overloaded with sentiment recognition resources. It comprises parsers, lexicon parts of speech (POS) taggers, and many NLP (natural language processing) libraries and other tools. Although the central part of current existing sentiment analysis models is designed for the English dialect [2], the growth of social media traffic in resource-poor dialects other than English has encouraged the development of many non-English sentiment recognition applications. The sentiment classification in idioms such as

✉ Hsien-Tsung Chang
smallpig@widelab.org

[1] Department of Computer Science, IBADAT International University Islamabad, Pakpattan Campus, Street, Pakpattan 57400, Punjab, Pakistan

[2] Centre for Lifelong Learning, Universiti Brunei Darussalam, Gadong BE1410, Brunei

[3] Bachelor Program in Artificial Intelligence, Chang Gung University, No. 259, Wenhua 1st Rd., Guishan Dist., Taoyuan 33302, Taiwan, ROC

[4] Department of Computer Science and Information Engineering, Chang Gung University, No. 259, Wenhua 1st Rd., Guishan Dist., Taoyuan 33302, Taiwan, ROC

[5] Department of Physical Medicine and Rehabilitation, Chang Gung Memorial Hospital, No. 5, Fusing St., Gueishan Dist., Taoyuan 33301, Taiwan, ROC

[6] Computer Science and Information Systems Department, Applied Computer Science College, King Saud University, Riyadh 12571, Saudi Arabia

Urdu is equally essential for the functioning sentiment classification tools and sentiment recognition development models [3]. Almost 250 million people speak and understand Urdu in Pakistan and India.

The absence of acknowledged lexicon resources for the Urdu language makes sentiment analysis in Urdu literature challenging [4, 5]. Regarding NLP tools, libraries, and famous sentiment lexicons like SentiWordNet, the English language has a wealth of resources. However, regarding linguistic resources and tools, like the need for well-known dictionaries, the Urdu language is regarded as one of the resource-poor languages.

Models based on deep learning have shown better performances for various fields such as emotion classification [6–9], medical data classification [10], and intelligent gaming [11] projects but on the other hand Urdu Language still considered resource deprived due to shortage of datasets, and other resources. Although Recently, few research scholars have start to shown bit of interest to solve basic tasks related to Urdu sentiment analysis [12–14] and emotion classification [15] but still huge amount of work needs to be done. The existing literature on Urdu sentiment analysis mainly has two approaches: conventional machine learning-based models and lexicon-based approaches. Conventional ML models include SVM, naive Bayes, and KNN on Urdu dialect features such as parts of speech tags and N-Gram. This approach has two main problems: (I) the feature generation or engineering process is tedious, time-consuming, and laborious; (II) the feature space where the algorithm to be trained becomes narrow and very high-dimensional. Therefore, the performance decreases. On the other hand, lexicon-based techniques need to use and create a pool or database of words and their related sentiments.

Recently, word embedding with deep neural networks (DNN) has been proposed to address the above-mentioned traditional ML learning issues and used in many recent research studies [13, 14]. Word embedding, is a numeric vector number increase the model performances, that considers various lexical associations and is produced using a neural language model [16–18].

DNN architectures such as recurrent neural networks (RNN) and convolutional neural networks (CNN), mainly focusing on NLP-related tasks, were commonly used for learning word embedding, clustering on learned feature vectors and text classification [19]. The main reason for the popularity of these models is that CNN is very efficient at local feature learning, and RNN has become very popular for sequential data processing because of its structure. Although RNNs are very good for many NLP-related tasks, they need help capturing long-term data because of exploding and vanishing gradient problems. Such long-term deficiencies are widespread in all NLP-related tasks.

Special type of RNN models such as LSTM [20] GRU [21] were designed to address the long-term dependencies problems. LSTM and GRU got attention from many researchers to solve NLP-related tasks. Later, Bi-directional LSTM and GRU were introduced to capture the long term dependencies in forward and backward in both directions. By combining the back and forward hidden layers, these particular types of RNN show state-of-the-art performances on sequential data.

Although BI-GRU and BI-LSTM have been commonly employed to solve NLP-related problems but both model have two main issues. First, LSTM and GRU cannot focus on essential words to capture crucial contextual information. Second, large-dimensional input space problems in NLP-related tasks are common; these issues add to the model's complexity and make optimization extremely challenging. Many methods were proposed to solve these issues. CNN was used to reduce the feature space dimensionality and extract useful information from features [22], and attention mechanisms were used to focus on essential words and give high wattage to important words [23].

However, current sentiment analysis techniques typically address a few issues while ignoring others. For example, [24] used the BERT pre-trained word embedding model with LSTM to extract semantics and sentiment for emotion classification. However, their proposed model ignores the need to focus on important words of a sentence or important part of text. Similarly, Study [25] combined CNN and BI-LSTM, using an attention method to highlight key phrases while ignoring the co-occurrence of long and short dependencies in their model. Pre-trained word embeddings with CNN were used by the authors of Study [26] to increase performance, but they should have captured long-term dependencies and concentrated on essential words as well.

Our proposed approach initially uses pre-trained word-to-vector models such as BERT and fastText to capture contextual information. CNN is used for local feature extraction, while Bi-LSTM employed to captures long-term dependencies in both directions. Finally, an attention technique is used to make the model capable of distinguishing between important and less important words. We performed multiple experiments on two publicly accessible Urdu datasets to demonstrate the effectiveness of our proposed model.

The following are the primary contributions of this paper:

- Building a novel attention-based deep Urdu sentiment analysis model
- Using our proposed architecture, we determine the effectiveness of context-based pre-trained (BERT) and fixed pre-trained (fastText) embedding models.
- Effectiveness of an Additional Layer of Bi-LSTM
- Extensive experiments on two publicly available datasets revealed that our proposed algorithm achieves better

results than existing Urdu language architectures [12, 14].

The rest of the papers are structured: “Research objectives and motivation” section represents research objectives and motivation; “Literature review” section describes literature review; and “Materials and proposed methodology” section contains the proposed methodology and used datasets. “Experiments and results” section explains experiments and the achieved results. Finally, “Conclusion and future work” section represents the conclusion and future work.

Research objectives and motivation

Nowadays, with a considerable amount of textual data generated using mobile phones, every person is inquisitive about product and service reviews and political sentiments. Even though more than 250 million people in South Asia communicate in Urdu, Urdu is the national language of Pakistan. Urdu is widely spoken and written in many union territories and states in India, but still, Urdu is considered one of the most resource-deprived languages. Therefore, the SA of Urdu language is equally worth it. This paper’s primary research goals are to create and evaluate a hybrid model for Urdu sentiment analysis that combines mBERT,CNN, Bi-LSTM, and with an attention mechanism. Furthermore, the research utilizes extensive classifiers to assess and classify text sentiment proficiently. Additionally, the study explores the impact of Bi-LSTM layers on performance.

Literature review

In the last few years, SA has become one of the hottest topics for researchers. In this section, we looked at some of the methodologies used in the literature to classify textual data written in Urdu.

A linguistically driven sentiment recognition system was employed in a study [27] on the morphologically complicated and rich Urdu dialect. Urdu is morphologically rich and complicated, and suppleness in grammatical rules necessitates an improved or entirely new way. As a result, authors concentrate on identifying SentiUnits rather than subjective terms in the given text. They used shallow parsing chunking to link SentiUnits to their targets. They build an annotated sentiment lexicon of Urdu words for their model training. Each lexicon is labeled with a sentiment, such as negative or positive. Two datasets were created of user comments from electronic appliances and movies. The algorithm performance demonstrates that their stated model achieves the best results in sentiment detection of Urdu text.

The supervised machine learning approach and the lexicon-based methods for sentiment detection of Urdu text are contrasted in a study [28]. The results demonstrate that lexicon-based classifiers outperform conventional ML models approaches such as KNN, SVM, and DT regarding accuracy. Both approaches tested on corpora that three human experts for supervised learning had manually labeled. A total of 6025 Urdu language comments from 151 Urdu websites and blogs under 14 distinct domains were gathered from various sources. In the first step of the lexicon-based technique, Urdu lexicons are generated, and in the second step, an Urdu lexicon classifier is developed for classification. Negative and positive words were collected from diverse domains to form the Urdu vocabulary. Negative words go into one file, while positive words go into another. An Urdu tagger is used to assign parts of speech (POS) to each word. There were 9578 positive and 11,739 negative lexicons in the newly constructed Urdu sentiment lexicon and separate files for intensifiers, context-dependent terms, and negations. The rule-based algorithms were created utilizing a newly constructed dataset of 6025 comments and regularly used samples from everyday life. Lexicon-based Urdu sentiment analysis’s performances fall short of expectations. The poor results of lexicon-based or rule-based Urdu sentiment analysis can be attributed to various factors. The need for widely used Urdu Lexicon databases is the first significant obstacle for rule-based or lexicon-based techniques. Sundered lexicon databases can play a significant role in lexicon/rule-based techniques to enhance performance. Another critical factor is the inability of rule-based or lexicon-based techniques to capture long-term dependencies or concentrate on the semantic meanings of words or sentences. The lexicon- or rule-based techniques could have identified mockery sentences. If a sentence has more negative terms, rule-based techniques will classify it as negative without considering the statement’s overall semantics. Similarly, a sentence’s classification will consider the number of positive terms it includes.

In the study [29], Asghar et al. designed a lexicon-based algorithm for Urdu SA. The authors suggested a word-by-word translation strategy for developing an inclusive Urdu lexicon using existing resources. Modifiers are also gathered, categorized, and given the appropriate polarity scores. The findings indicate that this methodology’s polarity scores are more accurate than those given by baseline approaches.

Traditional ML techniques, including SVM, KNN, RF, and DT, have been utilized in [30]. The study’s authors gathered Urdu text data from 151 internet blogs of 14 domains. All sentences are categorized as positive, negative, or neutral. Some pre-processing steps were applied, such as normalization, tokenization, and stop-word removal. The authors then used machine-learning techniques to categorize statements as positive, neutral, or negative. The finest classifiers for classifying sentiment of Urdu text were IBK,J48, and Lib

SVM. In another study, Mukhtar et al. [28] employed supervised machine learning approaches to classify sentences. The authors used KNN SVM, and DT algorithms for classification. According to the results, KNN beats the other algorithms in terms of performance.

Recently, Khan et al. in study [12] collected 9601 user generated comments from different online social media websites such as X, Facebook, and YouTube. In the second phase, the authors manually annotated all these collected reviews with the help of three human experts. Out of 9601 reviews, 4843 were positive, and the remaining were negative comments. In the 3rd phase, authors take basic NLP pre-processing steps, such as steaming, removing stop words, and normalizing Urdu text. Various conventional ML algorithms such as RF, NB, SVM, Ada-Boost, MLP, and LR, were employed to categorize the Urdu text using various n-gram, uni-gram, bi-gram, tri-gram, and combinations of these features. The results show that the LR algorithm with the (2–3) combination feature outperformed the other algorithms.

Another study, [31], has been published in the domain of SA of Urdu text. One of the main contributions of their research study was the manually annotated dataset, which contains 6000 Urdu sentences. The proposed dataset was cleaned and pre-processed. After cleaning and pre-processing, authors have proposed multiple DL algorithms like stacked LSTM, Bi-LSTM with and without attention layers, and CNN. All these DL algorithms have been explored on the proposed dataset for the Urdu SA task. The authors have also compared the performance of different pre-trained fixed words to vector models such as fasText, glove, word2vec, and SAMAR embedding. Results revealed that BI-LSTM with attention layers using SAMAR embedding outperformed all other used models. BI-LSTM with attention layer with SAMAR embedding achieved the highest accuracy of 77.92%. The following figure represents their overall proposed model.

Recently, models based on transformer learning showed better performances on various tasks related to NLP. The authors of the study [14] classified Urdu sentimental text by implementing a set of machines and deep learning approaches. Making a corpus for sentiment analysis was the study’s main contribution. The authors gathered user reviews for this project from different Social media networks including Facebook, Instagram, YouTube, X. These user evaluations in Urdu cover various topics, including sports, entertainment, software, mobile devices, politics, and movies. Two human specialists painstakingly classified each review as positive, negative, or neutral. A third human annotator arbitrated the dispute between the two human annotators. In total, The corpus comprises 9312 reviews. Various machine, rule-based, and deep learning models with n-gram features are applied. The rule-based model had the worst performance out of all the models. The SVM model beat all other machine learning models. FasText and BERT embedding used various DL algorithms, including BI-LSTM, LSTM, BI-GRU, GRU, and multilingual BERT. The transformer based large language model mBERT beats all other experimented algorithms.

In addition to the above approaches, attention-based mechanisms are becoming popular for English-language sentiment analysis, which can also be useful for Urdu SA. MeiZhen Liu et al. [32] presented a co-attention network-based mechanism that belongs to a 1-pair hop (for more attention) and an interactive mechanism (highlights the significant features). This model calculates bi-directional attention weight. Yongxue Shan et al. [33] designed a bi-graph attention network (BiGAT) with a retrieval-based attention mechanism and aspect-specific mask operation. The attention graph network collects neighbor information and removes useless information or noise.

Eniafe Festus Ayetiran et al. [34] proposed an attention-based technique consisting mainly of a Bi-LSTM and convolutional layers. They use CNN for feature extraction. Hossein Sadar et al. [35] paper proposes an attention-Based CNN with Transfer Learning, named as ACNN-TL, that combines attention mechanisms with transfer learning techniques. Language models are used as the backbone to record the characteristics of words and sentences. There is a significant improvement in accuracy due to contextual representations and transfer learning. Lixin Zhou et al. [36] presented a two-attention-based novel architecture. This model collects and investigates emoji information from different perspectives and then puts certain information into Bi-LSTM to obtain the five big personality traits. Aziz Khan F Pathan et al. [37] proposed an attention-based "position-aware Bi-LSTM network (attention-based multi-model contextual fusion strategy) for aspect-based opinion mining and lexicon approach. Here, the lexicon is used as a sentiment intensity score, which weights the sum of all information. Then, this lexicon sentiment intensity is loaded into Bi-LSTM. Mahesh G. Huddar et al. [38] investigated utterances and proposed an attention-based multi-model contextual fusion strategy with Bi-LSTM. First, they perform a fusion operation and extract important contextual information from the utterance. IEMOCAP (for emotions) and CMU-MOSI (for sentiment) were used.

Due to the poor “performance of multi-model sentiment classification and multihop grammatical distance, Ashima Yadav et al. in study [39] proposed a model known as “Deep Multilevel Attentive Network (DMLANet).” This model builds a bridge between two modalities, like text and image. For semantics and self-attention, they used a Bi-attentive visual map with CNN,” which automatically classifies real-world datasets like “VSA single and MVSA multiple. Mohammad Ehsan Basiri et al. [23] presented an “attention-based Bidirectional CNN-RNN Deep Model (ABCDM)” that takes out past and future contexts using BI-

Table 1 Statistics of experimental datasets

Sentiment	UCSA-21	UCSA
Negative	4758	2787
Positive	4843	3422
Neutral	–	3103
Total reviews	9601	9312

LSTM and GRU layers. This model also uses convolution, pooling, and attention mechanisms. For experiments, to asses their model performance authors used five review datasets and three Twitter datasets.

Materials and proposed methodology

Datasets

Our Paper used two Urdu datasets, UCSA-21 [12] and UCSA [14], to validate our proposed model. We can tell from the statistics of the datasets that both Urdu corpora are well-balanced with each classification class. Table 1 describes the datasets.

Proposed model

The proposed model provides a new attention-based DL algorithm for polarity detection of user comments to overcome issues with existing deep learning models for SA. In this proposed algorithm, fastText and BERT pre-trained and transformer-based word embedding, two-layer Bi-LSTM, attention method, and CNN are used to improve vector representation and capture local features and long-term dependencies. To enable the suggested approach to obtain context-based vector representations, we used fastText and multilingual-BERT. The output of the convolutional layer(CL) After local features extraction from the CL is then applied with two stacked layers of Bi-LSTM to capture the long-term dependencies in both backward and forward directions. Then, attention technique is applied to emphasis on important words and ignore less important or repetitive words. Then finally after FC layer, classification function such as soft-max is used for final classification.

Input layer

Input layer is the initial layer of the proposed algorithm. Used Datasets are mentioned in Table 1 are inputted to proposed model for training and testing purposes.

Urdu pre-processing steps

The primary aim of pre-Processing steps is to increase the model performance and reduce its computational cost. As in Other languages, Urdu text pre-processing is equally essential for Urdu language for better and more efficient results. Because of Urdu language complex morphology structure, and due to lack of available resources make Urdu language pre-processing more difficult and complex. Tokenization, steaming, normalization, and stop word removal steps were applied before model training. these pre-processing steps are explained in details in study [12, 14].

Embedding layer

mBERT [40] known as Multi-linguagle BERT, based on famous BERT large language models, pre-trained on 104 dialects together with Urdu. Another fixed embedding model known as fastText [41], has been trained on 157 variouse dialects together with Urdu language. we are not going to explain these embedding models because these techniques are well known and too obvious. There are two basic reseasons to choosing these two techniques: 1) both of the techniques are pre-trained on Urdu language. 2) compare fixed pre-trained (fastText) word embedding with context based (mBERT) embedding model for Urdu language.

Feature extraction with CNN

CNN is well known for local feature extraction, is being widely used for local feature extraction in sentiment classification [13, 23] related tasks. Let's symbolically denotes the $K-$dimensional vector corresponding to the ith token in a user review as $x_i \in R^k$ where k denotes the dimensionality of the vector. Considering a total size or length of n tokens in the review, we can concatenate these individual token vectors to form a sequence. Mathematically, this can be illustrated as follows in Eq. 1.

$$X_{1:n} = x_1 + x_2 + x_3 \cdots + x_n \tag{1}$$

In Eq. 1, the + denotes concatenation process. Consider a scenario where x_i, x_{i+1}, x_{i+2}, $\ldots x_{i+j}$ represent a continuous sequence and are identical to $x_{i:i+j}$. Suppose W$\in R^{hk}$ represents the convolutional filters. To construct a new feature matrix, we apply convolutional filters to an n X K-dimensional phrase matrix, where n represents the total number of phrases and K denotes the dimensionality of each phrase vector. This convolutional operation is performed with a window size of h. The phrase vector $x_{i:i+j}$, encapsulates the local feature information from the ith to the $(i+j)th$ position. For each window of words, the segment $x_{i:i+h-1}$extracted from Eq. 2 is utilized to generate a feature

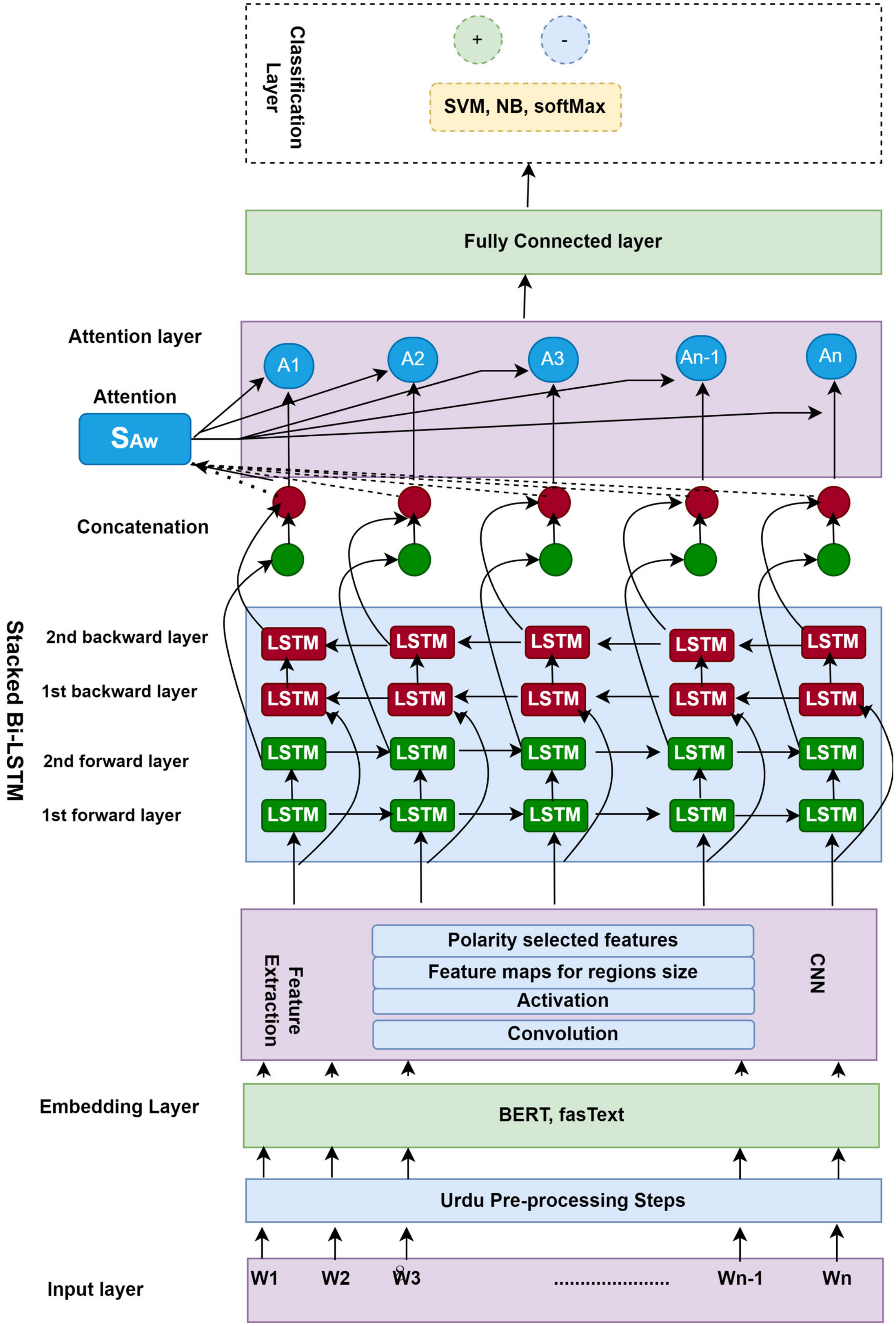


Fig. 1 An attention-based proposed architecture for Urdu sentiment analysis

c_i.

$$C_i = f(W.x_{i:i+h-1} + b) \tag{2}$$

The variable b represents the bias term and belongs to the set of real numbers. Additionally, f denotes an activation function. To produce a feature map, the convolutional filter is applied to each window of words using Eq. 3.

$$C = [C_1, C_2, C_3, C_{n-h+1}] \tag{3}$$

Let $C \in R^{n-h+1}$,The process of creating a single feature map from a convolutional filter has been explained above. When employing multiple m filters in a convolutional layer, the result will be $m(n-h+1)$ features. Since feature selection can influence long-term dependencies, especially in the early LSTM layers, the max pooling layer is omitted for feature maps. Instead, the features are directly fed into the Bi-LSTM layer before proceeding to the fully connected layer to capture long-term dependencies.

Two layer stacked bi-LSTM

LSTM, a distinctive variant of recurrent neural networks (RNNs), is engineered to address the challenge of capturing long-term dependencies and mitigating issues like vanishing or exploding gradients encountered by conventional neural networks. The proposed model contains two layer stacked BI-LSTM to capture forwarded and backward dependencies. An extra BI-LSTM layer is incorporated to enhance the model's overall performance.

The sequence of input such as x_1, x_2, $x_3 \ldots x_t$ enters the Bi-LSTM first hidden layer for time t. It travels through two routes: the forward route (a_1, x_2, $a_3 \ldots a_t$ is utilized to gather contextual information from all preceding time steps,while the reverse route (c_1, c_2, $c_3 \ldots c_t$ is employed to aggregate information from all incoming time steps. Finally, all output layers are integrated and fed to the attention layer.

In Fig. 1, each LSTM [20] cell has a memory cell U_t with three gates named as a forget gate f_t, an input gate i_t, an output gate o_t. Due to these gates, the LSTM network minimizes gradient vanishing problems. The memory cell is used to memorize values over different time sequences, and the gates regulate the flow of information into and out of the memory cell. Mathematically, the relationship between the different gates, hidden states, and memory cells can be formally described as follows for the initial forward direction layer and hidden statea_t.

$$\begin{aligned}
i_t^a &= \sigma(U_i^a x_t + W_i^a a_(t-1) + b_i^a) && (4)\\
f_t^a &= \sigma(U_f^a x_t + W_f^a a_(t-1) + b_f^a) && (5)\\
o_t^a &= \sigma(U_o^a x_t + W_0^a a_(t-1) + b_o^a) && (6)\\
U_t^a &= \tanh(U_u^a x_t + W_u^a a_(t-1) + b_u^a) && (7)\\
C_t^a &= (i_t^a * U_t^a * f_t^a C_(t-1)^a) && (8)\\
a_t &= o_t^a * tanh C_t^a && (9)
\end{aligned}$$

The following equations are utilized to compute the hidden state b_t for the second forward layer:

$$\begin{aligned}
i_t^b &= \sigma(U_i^b x_t + W_i^b b_(t-1) + b_i^b) && (10)\\
f_t^b &= \sigma(U_f^b x_t + W_f^b b_(t-1) + b_f^b) && (11)\\
o_t^b &= (U_o^b x_t + W_0^b b_(t-1) + b_o^b) && (12)\\
U_t^b &= \tanh(U_u^b x_t + W_u^b b_(t-1) + b_u^b) && (13)\\
C_t^b &= (i_t^b * U_t^b * f_t^b C_(t-1)^b) && (14)\\
b_t &= o_t^b * \tanh C_t^b && (15)
\end{aligned}$$

The formal mathematical relations for the first backward direction layer, the hidden state c_t are provided below.

$$\begin{aligned}
i_t^c &= \sigma(U_i^c x_t + W_i^c c_(t-1) + b_i^c) && (16)\\
f_t^c &= \sigma(U_f^c x_t + W_f^c c_(t-1) + b_f^c) && (17)\\
o_t^c &= \sigma(U_o^c x_t + W_0^c c_(t-1) + b_o^c) && (18)\\
U_t^c &= \tanh(U_u^c x_t + W_u^c c_(t-1) + b_u^c) && (19)\\
C_t^c &= (i_t^c * U_t^c * f_t^c C_(t-1)^c) && (20)\\
c_t &= o_t^c * \tanh C_t^c && (21)
\end{aligned}$$

The formal mathematical relations for the 2nd backward direction layer, the hidden state d_t, are provided below.

$$\begin{aligned}
i_t^d &= \sigma(U_i^d x_t + W_i^d d_(t-1) + b_i^d) && (22)\\
f_t^d &= \sigma(U_f^d x_t + W_f^d d_(t-1) + b_f^d) && (23)\\
o_t^a &= \sigma(U_o^a x_t + W_0^a d_(t-1) + b_o^d) && (24)\\
U_t^d &= \tanh(U_u^d x_t + W_u^d d_(t-1) + b_u^d) && (25)\\
C_t^d &= (i_t^d * U_t^d * f_t^d C_(t-1)^d) && (26)\\
d_t &= o_t^d * \tanh C_t^d && (27)
\end{aligned}$$

Attention mechanism

In the existing literature, attention techniques were suggested to circumvent some of the interpretations executed by pooling operators. When we read a sentence, we can recognize the most essential portions in context and ignore repetitive or misleading information. Attention mechanism is used to tackle this tricky situation for NLP related tasks by giving more weights to important part of the sentence and giving less weights to less important words. A weighted combination (see Fig. 2) of all hidden states is a common approach

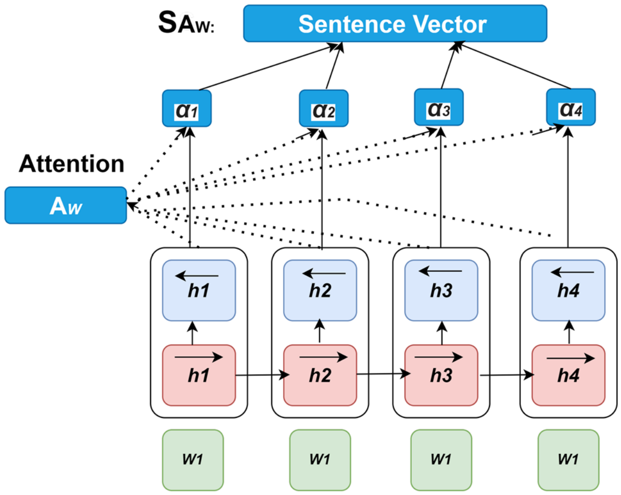


Fig. 2 Attention mechanism adopted from study [44]

to assigning numerous weights to each word in a sentence. An attention sentence encoder [42, 43] applies various values and weights to the phrase's own words, translating the hidden states of a complete sentence into a vector illustration.

$$\alpha_t = \frac{exp(VT.h^-)}{\sum_t exp(V.h^-)} \tag{28}$$

$$S_{Aw} = \sum_t \alpha_t h_t \tag{29}$$

The symbols h and h^- are forwarded and backward hidden states from Bi-LSTM, and V represents a trainable parameter.

S_{Aw} is the average of assigned weights to lSTM hidden states as shown in the Fig. 2.

$$\overrightarrow{h_{tLSTM}} = \overrightarrow{(LSTM)}(c_t), \quad t \in (1, m) \tag{30}$$

$$\overleftarrow{h_{tLSTM}} = \overleftarrow{(LSTM)}(c_t), \quad t \in (m, 1) \tag{31}$$

We may now get an annotation for each word, W_t, by concatenating backward and forward contexts in the equation.

$$h_{tLSTM} = \overrightarrow{h_{tLSTM}} \oplus \overleftarrow{h_{tLSTM}} \tag{32}$$

The attention approach is applied to the h_{tLSTM} model to enable it to pay more or less attention to various terms inside the comment. On h_{tLSTM}, the attention technique enables the model to give particular terms in the sentence more or less attention. In order to accomplish this, we updated the feature vector by extracting informative words in the comment as follows shown in Eq. 33.

$$u_{tLSTM} = \tanh((W_{wLSTM} \cdot h_{tLSTM}) + b_{wLSTM}) \tag{33}$$

$$\alpha_{tLSTM} = \frac{exp(u_t^{T_{LSTM}}) \cdot u_{wLSTM}}{\sum_t exp(u_t^{T_{LSTM}} \cdot u_{wLSTM})} \tag{34}$$

$$S_{LSTM} = \sum_t (\alpha_{tLSTM} \cdot h_{tLSTM}) \tag{35}$$

Where u_t represents the hidden description of h_t and u_w represents a context vector that is learned and initialized during training time. The relevance of the word u_t is premeditated by equating u_t to u_w and then normalizing it as given in Eq. 34. Finally, these critical values and weights α_t are integrated into S via a weighted sum as shown in Eq. 35. S is the sentence vector, which summarizes the information about words in the sentence. The fully linked layer is then fed the phrase vector S. After that, classification algorithms like softmax, SVM, NB, and KNN are passed to the output of the fully connected layer.

Experiments and results

This section of the paper contains the evaluation criteria used to evaluate the proposed model, The outcomes of the conducted experiments, the effectiveness of each model, and discussion on these results in light of the study's objective.

Evaluation measures

Accuracy (36), precision (37), recall (38), and F1-measure (39) are being extensively used to assess the performances of the models developed for task like text classification. Following mathematical equation are used to calculate the evaluation metrics.

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative} \tag{36}$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{37}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{38}$$

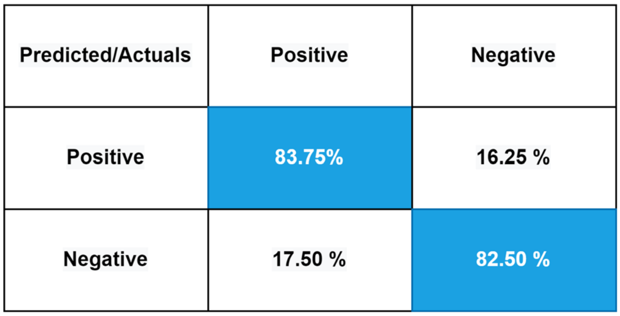
$$F1 - score = \frac{2(Precision)(Recall)}{Precision + Recall} \tag{39}$$

Hyper parameter settings

The Keras library used written in python computer language to implements our attention-based suggested model using multiple-word embedding techniques. The datasets are segmented into training (80%) and testing (20%) parts. The embedding layer uses weights from publicly accessible transformer-based multilingual BERT and pre-trained fastText word-to-vector models. mBERT has undergone pre-training across more than 104 languages togethor with Urdu. fastText is similarly pre-trained on Urdu text data, utilizing various data sources. The BERT model is utilized as the foundation, with 12 transformer levels and 768 hidden layers. We utilize a two-layer stacked Bi-LSTM with 128 memory units in the sequential layer. Filters of 4 and 32 are used, with kernel sizes of 4 and 6 utilized in the convolutional layer. A sigmoid activation function is employed as the activation function. Due to hardware constraints the training batch size is set to 32 with a 0.2 dropout rate. The additive attention scoring function is used at attention layer to calculate the score. It allows the attention mechanism to learn more complex patterns and interactions between the query and key vectors. The network underwent training via the back-propagation method, leveraging the Adam stochastic optimizer. The binary cross-entropy loss function is utilized, it is extensively being used for binary classification problems where each instance can belong to one of two classes. The training method employs five-fold cross-validation to prevent over-fitting.

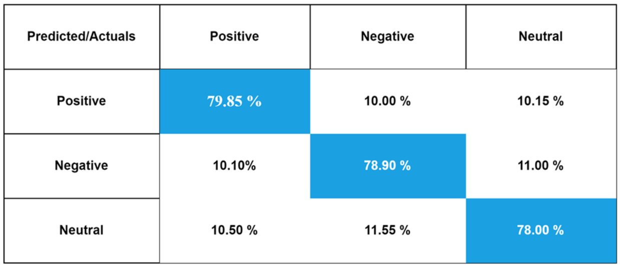
Results

The confusion matrix is a metric for determining whether or not a classification is valid. The proposed attention-based model's confusion matrix for the UCSA-21 and UCSA is shown in Figs. 3 and 4, respectively. Out of all, 83.75% of positive reviews are correctly classified as positive, and only 16.25% of positive comments are mistakenly recognized as negative due to manual annotation and issues related to the morphological structure of Urdu text. On the other hand, only 17.50% of negative user comments are wrongly classified as positive reviews, while 82.50% of negative reviews are classified correctly against the UCSA-21 data corpus. Similarly, in Fig. 4, Only 10.00% of positive comments were mistakenly classified as negative and 10.15% as neutral, while 79.85% of positive comments were identified correctly as positive. Only 10.10% and 11.00% of negative comments were erroneously labeled as positive and neutral, respectively, while 76.00% of negative comments were acceptably recognized as negative. Using attention-based Urdu sentiment analysis on the UCSA dataset, just 10.50% and 11.55% of neutral comments were misclassified as positive and negative, respectively, while 78.00% of neutral comments were identified correctly as neutral.



Predicted/Actuals	Positive	Negative
Positive	83.75%	16.25 %
Negative	17.50 %	82.50 %

Fig. 3 Confusion matrix of an attention-based proposed model against UCSA-21 Corpus



Predicted/Actuals	Positive	Negative	Neutral
Positive	79.85 %	10.00 %	10.15 %
Negative	10.10%	78.90 %	11.00 %
Neutral	10.50 %	11.55 %	78.00 %

Fig. 4 Confusion matrix of an attention-based proposed model against UCSA Corpus

Best embedding model and machine learning classifier

The quality of word embedding approaches ultimately influences the performance of the algorithm. To evaluate the effectiveness of our proposed attention-based architecture, we used two types of fixed and context-based word embedding models. fastText, a pre-trained fixed word-to-vector model, works similarly to word2vec and mBERT, a context-based word-to-vector model based on the transformer technique. The outcomes of our proposed architecture, which was applied to the UCSA and UCSA-21 datasets using mBERT and fastText, are shown in Tables 2 and 3, respectively. The results shows that, our proposed model performed better with mBERT word embedding than fasText. Due to context based embedding, the mBERT teachnique proved more effective as compared to fixed embedding. Similarly, SVM model perform better than other model such as KNN, NB and softmax as classification function after the FC layer. Regarding the achieved results in terms of F1-score, accuracy, precision and recall, the SVM model beat all other employed classifiers, which is consistent with study [12] and [14]. On UCSA and UCSA-21 corpus, the proposed CNN attention-based model using stacked two-layer Bi-LSTM achieved accuracy of 78.91% and 83.12%, respectively. However, employing the softmax function, the effectivness of our suggested architecture remained slightly lower. Similarly, our suggested algorithm performance with NB and KNN (K = 10) as classifier functions remained significantly superior to that of the softmax function. In Fig. 5, part (a) illustrates the UCSA dataset, while part (b) depicts the UCSA-21 dataset. The accuracy of fastText and mBERT embedding models is compared using SVM, NB, KNN, and Softmax classifiers.

Table 2 Results comparison of our proposed model using fastText vs mBERT, and one-layer-bi-LSTM vs two-layer-bi-LSTM against UCSA Corpus

Embedding	Classifier	LSTM layers	Accuracy %	Precision %	Recall %	F1-score %
fasText	SVM	1	77.85	76.10	78.00	77.03
		2	78.10	76.55	78.40	77.46
	NB	1	77.00	75.80	77.50	76.64
		2	77.35	76.35	77.65	77.09
	KNN	1	76.80	75.55	77.00	76.26
		2	77.10	76.05	77.45	76.74
	SoftMax	1	76.90	75.30	76.80	76.04
		2	77.05	75.75	77.20	76.46
mBERT	SVM	1	78.65	77.05	78.90	77.96
		2	**78.91**	**77.40**	**79.20**	**78.28**
	NB	1	78.05	76.80	78.45	77.61
		2	78.40	77.15	78.65	77.89
	KNN	1	77.95	76.45	78.00	77.21
		2	78.15	77.05	78.45	77.74
	SoftMax	1	77.50	76.30	77.80	77.04
		2	77.85	76.75	78.20	77.46

Values in bold represent the highest performance metrics achieved among all experimental settings

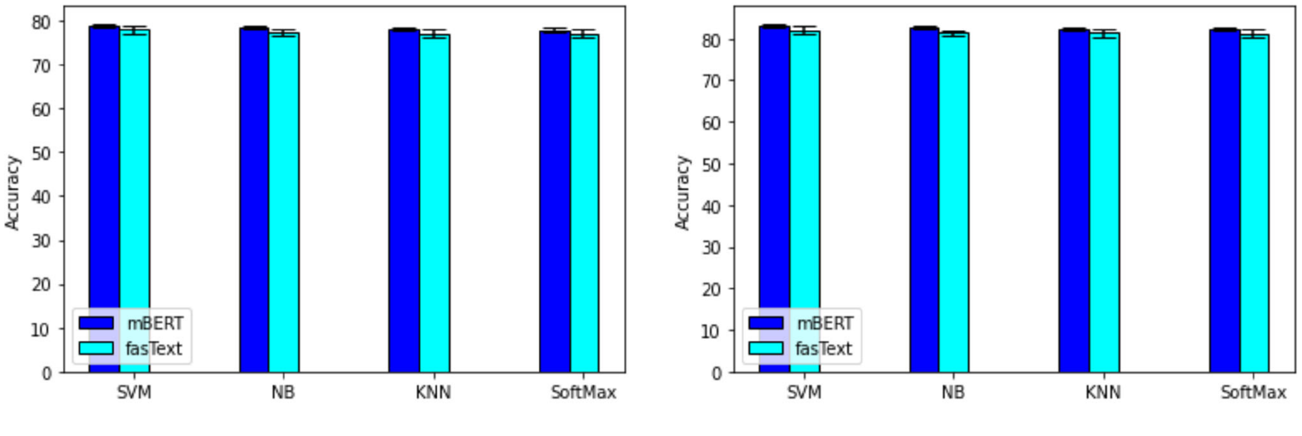
Table 3 Results comparison of our proposed model using fastText vs mBERT, and one-layer-bi-LSTM vs two-layer-bi-LSTM against UCSA-21 Corpus

Embedding	Classifier	LSTM layers	Accuracy %	Precision %	Recall %	F1-Score %
fasText	SVM	1	81.75	81.00	81.40	81.19
		2	81.95	81.25	81.60	81.42
	NB	1	81.00	80.80	80.15	80.47
		2	81.35	80.95	81.50	81.22
	KNN	1	81.20	80.50	81.05	80.77
		2	81.30	80.70	81.15	80.92
	SoftMax	1	80.90	80.25	80.65	80.44
		2	81.15	80.35	81.10	80.72
mBERT	SVM	1	82.75	82.00	82.45	82.22
		2	**83.12**	**82.20**	**82.70**	**82.44**
	NB	1	82.30	81.70	82.15	81.92
		2	82.55	81.85	82.50	82.17
	KNN	1	82.00	81.45	82.00	81.72
		2	82.25	81.60	82.25	81.92
	SoftMax	1	81.95	81.30	81.70	81.49
		2	82.20	81.40	82.00	81.69

Values in bold represent the highest performance metrics achieved among all experimental settings

Improving proposed model using stacked bi-LSTM

On the other hand, Bi-LSTM is a deep model of a feed-forward and feed-back neural network. The number of Bi-LSTM layers influences the effectiveness of classification. We examined the performance of two stacked Bi-LSTM layers vs. one Bi-LSTM layer with 128 units in each LSTM layer in the purportedly attention-based Urdu sentiment analysis model. The results of this experiment are summarized in Tables 2 and 3. Revealed results shows that, two stacked Bi-LSTM layers are sligthly more effective than a single-layer Bi-LSTM. two stacked Bi-LSTM layers improved classification results on average by +0.50% in accuracy, 0.75% in precision, and 0.70% in recall. Similarly, our proposed model with two-layer Bi-LSTM for all examined datasets produced better results. Therefore, stacked Bi-LSTM layers are believed to be sufficient for building higher-order feature representations of Urdu phrases, enabling them to be more easily classified. Our proposed attention-based model gained accuracy, precision, recall, and a f1-score of 78.65%, 77.05%, 78.90%, and 77.96%, respectively, against our UCSA corpus using one layer Bi-LSTM. On the other hand Proposed model achieves slightly higher accuracy, precision, recall, and f1-score are 78.91%, 77.40%, 79.20%, and 78.28%,



(a)

(b)

Fig. 5 An accuracy comparison between fastText and mBERT embedding models against, **a** represent UCSA and **b** represents UCSA-21 datasets

respectively against UCSA corpus. Similarly,The suggested model achieved accuracy, precision, recall, and f1 scores of 83.12%, 82.00%, 82.45%, and 82.22% against the UCSA-21 datasets using a one-layer Bi-LSTM. On the other hand, the accuracy, precision, recall, and f1-score of two-layer Bi-LSTM on UCSA-21 datasets were somewhat improved, are 82.75%, 82.20%, 82.70%, and 82.44%, respectively. Figures 6 describe the performance of 1-layer Bi-LSTM vs the performance of 2-layer Bi-LSTM against SVM, NB, KNN and SoftMax classifiers. We can See clearly by adding a layer of Bi-LSTM enhanced the model's performance.

Table 4 compares our proposed model results with existing models against UCSA-21 and UCSA corpora. Our proposed model outperformed existing models regarding accuracy, precision, recall, and F1 score. Figure 7, part (a) represents training and testing accuracy, while part (b) describes training and testing loss against UCSA-21 corpus. Similarly, Fig. 8, part (a) represents the training accuracy while part (b) represent testing loses of our proposed model against UCSA datasets.

Conclusion and future work

Recently, in SA and text classification, word embedding algorithms overall, BERT, CNN, and Bi-LSTM algorithms, in particular, have been extensively employed. These models have a few flaws, but their performance can be improved. For Urdu sentiment analysis, a new attention-based using CNN two-layer-Bi-LSTM deep algorithm is proposed in this study. After applying basic Urdu text pre-processing steps, our model exploits publicly accessible, pre-trained fastText and mBERT word-to-vector models. In order to create distinct feature maps and compress the dimensionality of the feature space, these generated vectors from embedding models are then passed to a convolutional layer different kernel sizes. Additionally, using CNN in our proposed model aims to facilitate the extraction of local features. After local feature extraction with CNN, future and past contexts are extracted as semantic representations of the input text via utilizing a stacked two-layer Bi-LSTM. An attention layer is applied to the Bi-LSTM outputs to give certain words in a review more or less weight. The semantic representations become more informative as a result of this. The attention layer outputs are then fed to a FC layer. Finally, the output of the FC with the sigmoid function is fed into classic machine learning models like SVM, which classify user reviews as positive, negative, or neutral. Several experiments were carried out on two Urdu text datasets to assess the outcome of the proposed model. Various ML and DL models with diverse features and embedding strategies are used for performance evaluations. The results on these Urdu datasets reveal that our suggested model significantly outperforms competing existing ML and deep neural models. An additional layer of Bi-LSTM is stacked to increase the effectiveness of our proposed architecture. The effectiveness of our proposed architecture is increased even more by adding an extra-stacked Bi-LSTM layer. Finally, the presented two-layer Bi-LSTM model outperformed the single-layer Bi-LSTM model. The goal of this study was to analyze the sentiment of Urdu text at the sentence level. We intend to analyze the utility of our proposed model on tasks including aspect-level sentiment analysis, rating prediction, and document-level sentiment analysis in the future. The proposed approach is developed for Urdu dialects but might also be helpful with other resource dialects, like Arabic. We will work to improve results in the future by extracting other features like sentence, word, and character levels. We will combine those features after feature extraction. Future aspect-based and multi-model sentiment classification models are possible extensions of this Urdu

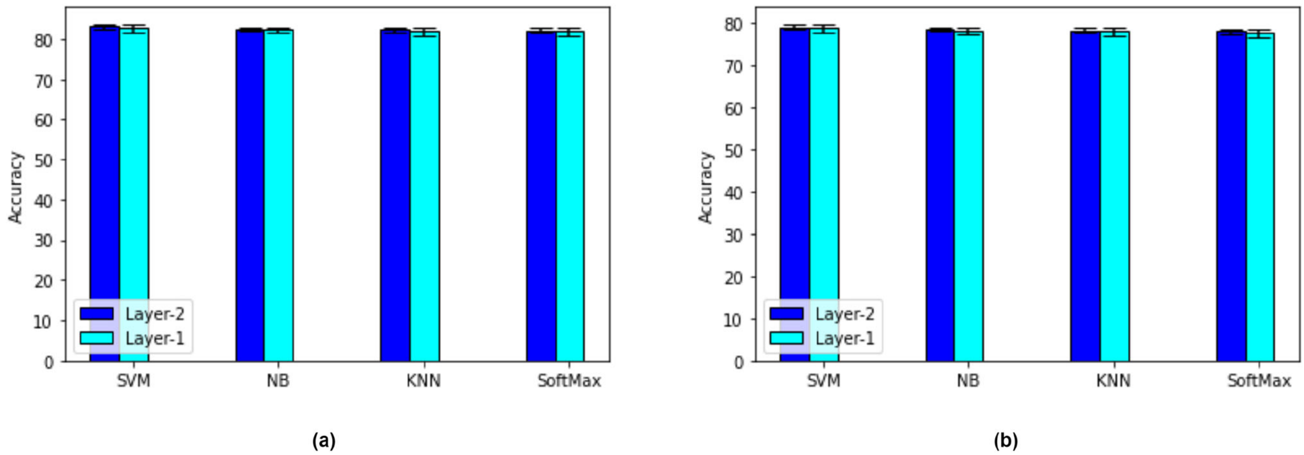


Fig. 6 An accuracy comparison of using 1-layer Bi-LSTM vs 2-layer Bi-LSTM, **a** represent UCSA-21 corpus and **b** represents UCSA Dataset

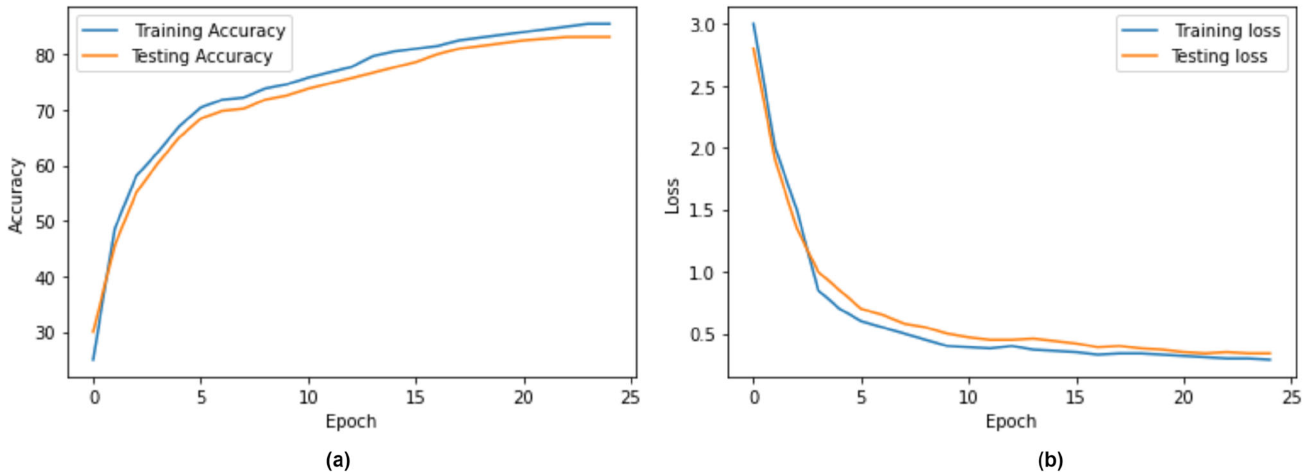


Fig. 7 Training and testing loss comparison against UCSA corpus

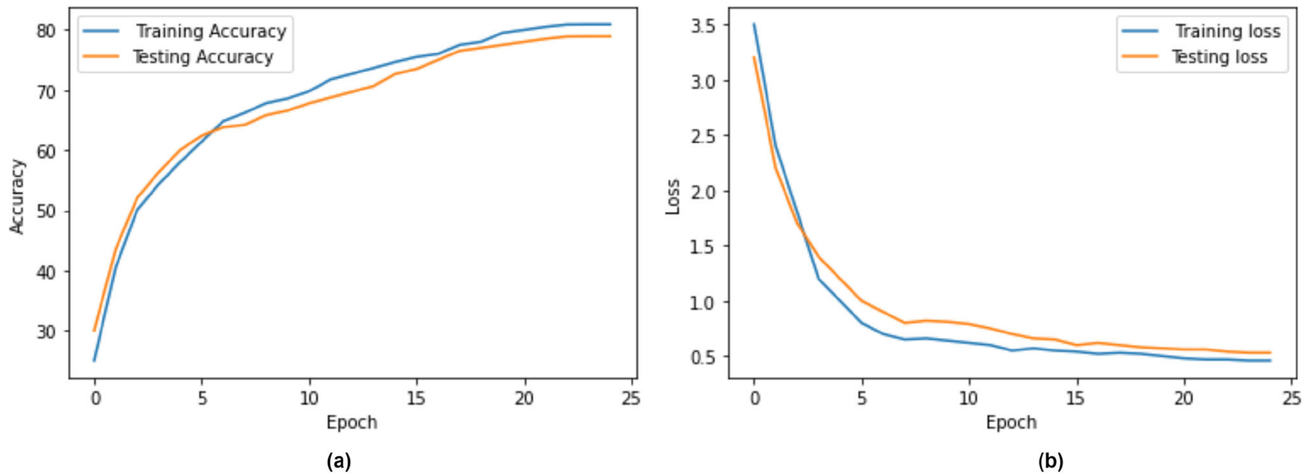


Fig. 8 Training and testing loss comparison against UCSA corpus

Table 4 Performance comparison with existing approaches

References	Dataset	Accuracy %	Precision %	Recall %	F1-Score %
[12]	UCSA-21	81.94	79.95	84.26	82.05
[14]	UCSA	77.61	76.15	78.25	77.18
	UCSA-21	82.50	81.35	81.65	81.49
Proposed Study	UCSA-21	**83.12**	**82.20**	**82.70**	**82.44**
	UCSA	**78.91**	**77.40**	**79.20**	**78.28**

Values in bold represent the highest performance metrics achieved among all experimental settings

text sentiment classification. Due to the fast growth of hybrid data on platforms like X, Snapchat, and Facebook, multi-model techniques are receiving increased attention from academic researchers; therefore, Multi-models are crucial for the future.

Funding Funding was provided by National Science and Technology Council (No. 112-2410-H-182 -026 -MY2), Chang Gung Memorial Hospital, Linkou (Nos. NERPD4N0231, BMRPA07) and King Saud University (No. RSP2024R458).

Data Availability All data used in this study is publicly available. The two Urdu sentiment analysis datasets used are: (1) UCSA-21 [12], available at IEEE DataPort (https://ieee-dataport.org/documents/urdu-corpus-sentiment-analysis), and (2) UCSA [14], accessible through GitHub (https://github.com/lalkhanbuzdar/Urdudataset/blob/main/Urdu%20Corpus%20for%20Sentiment%20Anlysis-20.rar). Both datasets can be downloaded freely for research purposes.

References

1. Ghorbanali A, Sohrabi MK (2023) A comprehensive survey on deep learning-based approaches for multimodal sentiment analysis. Artif Intell Rev 56:1479–1512
2. Wankhade M, Rao ACS, Kulkarni C (2022) A survey on sentiment analysis methods, applications, and challenges. Artif Intell Rev 55(7):5731–5780
3. Ullah F, Ullah I, Kolesnikova O (2022) Urdu named entity recognition with attention bi-LSTM-CRF model. In: Mexican international conference on artificial intelligence. Springer, pp 3–17
4. Kumar A, Jain AK (2022) Emotion detection in psychological texts by fine-tuning BERT using emotion-cause pair extraction. Int J Speech Technol 25(3):727–743
5. Mercha EM, Benbrahim H (2023) Machine learning and deep learning for sentiment analysis across languages: a survey. Neurocomputing 531:195–216
6. Amjad A, Khan L, Chang H-T (2021) Effect on speech emotion classification of a feature selection approach using a convolutional neural network. PeerJ Comput Sci 7:766
7. Amjad A, Khan L, Chang H-T (2021) Semi-natural and spontaneous speech recognition using deep neural networks with hybrid features unification. Processes 9(12):2286
8. Amjad A, Khan L, Chang H-T (2022) Data augmentation and deep neural networks for the classification of Pakistani racial speakers recognition. PeerJ Comput Sci 8:1053
9. Amjad A, Khan L, Ashraf N, Mahmood MB, Chang H-T (2022) Recognizing semi-natural and spontaneous speech emotions using deep neural networks. IEEE Access 10:37149–37163
10. Khan L, Shahreen M, Qazi A, Jamil Ahmed Shah S, Hussain S, Chang H-T (2024) Migraine headache (MH) classification using machine learning methods with data augmentation. Sci Rep 14(1):5180
11. Khan A, Shah AA, Khan L, Faheem MR, Naeem M, Chang HT (2024) Using vizdoom research platform scenarios for benchmarking reinforcement learning algorithms in first-person shooter games. IEEE Access, IEEE
12. Khan L, Amjad A, Ashraf N, Chang H-T, Gelbukh A (2021) Urdu sentiment analysis with deep learning methods. IEEE Access 9:97803–97812
13. Khan L, Amjad A, Afaq KM, Chang H-T (2022) Deep sentiment analysis using CNN-LSTM architecture of English and Roman Urdu text shared in social media. Appl Sci 12(5):2694
14. Khan L, Amjad A, Ashraf N, Chang H-T (2022) Multi-class sentiment analysis of Urdu text using multilingual BERT. Sci Rep 12(1):5436
15. Ashraf N, Khan L, Butt S, Chang H-T, Sidorov G, Gelbukh A (2022) Multi-label emotion classification of Urdu tweets. PeerJ Comput Sci 8:896
16. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
17. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
18. Rong X (2014) word2vec parameter learning explained. arXiv preprint arXiv:1411.2738
19. Li Q, Peng H, Li J, Xia C, Yang R, Sun L, Yu PS, He L (2022) A survey on text classification: from traditional to deep learning. ACM Trans Intell Syst Technol (TIST) 13(2):1–41
20. Graves A, Graves A (2012) Long short-term memory. Supervised Seq label Recurrent Neural Netw 37–45
21. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder–decoder for statistical machine translation. arXiv preprint arXiv:1406.1078

22. Mehta Y, Majumder N, Gelbukh A, Cambria E (2020) Recent trends in deep learning based personality detection. Artif Intell Rev 53:2313–2339
23. Basiri ME, Nemati S, Abdar M, Cambria E, Acharya UR (2021) ABCDM: an attention-based bidirectional CNN-RNN deep model for sentiment analysis. Futur Gener Comput Syst 115:279–294
24. Chatterjee A, Gupta U, Chinnakotla MK, Srikanth R, Galley M, Agrawal P (2019) Understanding emotions in text using deep learning and big data. Comput Hum Behav 93:309–317
25. Liu G, Guo J (2019) Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing 337:325–338
26. Rezaeinia SM, Rahmani R, Ghodsi A, Veisi H (2019) Sentiment analysis based on improved pre-trained word embeddings. Expert Syst Appl 117:139–147
27. Syed AZ, Aslam M, Martinez-Enriquez AM (2014) Associating targets with sentiunits: a step forward in sentiment analysis of Urdu text. Artif Intell Rev 41:535–561
28. Mukhtar N, Khan MA (2018) Urdu sentiment analysis using supervised machine learning approach. Int J Pattern Recognit Artif Intell 32(02):1851001
29. Asghar MZ, Sattar A, Khan A, Ali A, Masud Kundi F, Ahmad S (2019) Creating sentiment lexicon for sentiment analysis in Urdu: the case of a resource-poor language. Expert Syst 36(3):12397
30. Mukhtar N, Khan MA, Chiragh N (2017) Effective use of evaluation measures for the validation of best classifier in Urdu sentiment analysis. Cogn Comput 9:446–456
31. Naqvi U, Majid A, Abbas SA (2021) UTSA: Urdu text sentiment analysis using deep learning methods. IEEE Access 9:114085–114094
32. Liu M, Zhou F, Chen K, Zhao Y (2021) Co-attention networks based on aspect and context for aspect-level sentiment analysis. Knowl-Based Syst 217:106810
33. Shan Y, Che C, Wei X, Wang X, Zhu Y, Jin B (2022) Bi-graph attention network for aspect category sentiment classification. Knowl-Based Syst 258:109972
34. Ayetiran EF (2022) Attention-based aspect sentiment classification using enhanced learning through CNN-BILSTM networks. Knowl-Based Syst 252:109409
35. Sadr H, Nazari Soleimandarabi M (2022) ACNN-TL: attention-based convolutional neural network coupling with transfer learning and contextualized word representation for enhancing the performance of sentiment classification. J Supercomput 78(7):10149–10175
36. Zhou L, Zhang Z, Zhao L, Yang P (2022) Attention-based BILSTM models for personality recognition from user-generated content. Inf Sci 596:460–471
37. Pathan AF, Prakash C (2022) Attention-based position-aware framework for aspect-based opinion mining using bidirectional long short-term memory. J King Saud Univ-Comput Inf Sci 34(10):8716–8726
38. Huddar MG, Sannakki SS, Rajpurohit VS (2021) Attention-based multimodal contextual fusion for sentiment and emotion classification using bidirectional LSTM. Multimed Tools Appl 80:13059–13076
39. Yadav A, Vishwakarma DK (2023) A deep multi-level attentive network for multimodal sentiment analysis. ACM Trans Multimed Comput Commun Appl 19(1):1–19
40. Pires T, Schlinger E, Garrette D (2019) How multilingual is multilingual BERT? arXiv preprint arXiv:1906.01502
41. Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Ling 5:135–146
42. Soydaner D (2022) Attention mechanism in neural networks: where it comes and where it goes. Neural Comput Appl 34(16):13371–13385
43. Khurana D, Koli A, Khatter K, Singh S (2023) Natural language processing: state of the art, current trends and challenges. Multimed Tools Appl 82(3):3713–3744
44. Sardelich M, Manandhar S (2018) Multimodal deep learning for short-term stock volatility prediction. arXiv preprint arXiv:1812.10479

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.