Article

Deep Sentiment Analysis Using CNN-LSTM Architecture of English and Roman Urdu Text Shared in Social Media

Lal Khan [1], Ammar Amjad [1], Kanwar Muhammad Afaq [2] and Hsien-Tsung Chang [1,3,4,5,*]

[1] Department of Computer Science and Information Engineering, Chang Gung University, Taoyuan 333, Taiwan; d0829003@cgu.edu.tw (L.K.); d0721011@cgu.edu.tw (A.A.)
[2] Department of Computer Science and Information Engineering, Asia University, Taichung 413, Taiwan; 108121032@live.asia.edu.tw
[3] Bachelor Program in Artificial Intelligence, Chang Gung University, Taoyuan 333, Taiwan
[4] Department of Physical Medicine and Rehabilitation, Chang Gung Memorial Hospital, Taoyuan 333, Taiwan
[5] Artificial Intelligence Research Center, Chang Gung University, Taoyuan 333, Taiwan
[*] Correspondence: smallpig@mail.cgu.edu.tw

Abstract: Sentiment analysis (SA) has been an active research subject in the domain of natural language processing due to its important functions in interpreting people's perspectives and drawing successful opinion-based judgments. On social media, Roman Urdu is one of the most extensively utilized dialects. Sentiment analysis of Roman Urdu is difficult due to its morphological complexities and varied dialects. The purpose of this paper is to evaluate the performance of various word embeddings for Roman Urdu and English dialects using the CNN-LSTM architecture with traditional machine learning classifiers. We introduce a novel deep learning architecture for Roman Urdu and English dialect SA based on two layers: LSTM for long-term dependency preservation and a one-layer CNN model for local feature extraction. To obtain the final classification, the feature maps learned by CNN and LSTM are fed to several machine learning classifiers. Various word embedding models support this concept. Extensive tests on four corpora show that the proposed model performs exceptionally well in Roman Urdu and English text sentiment classification, with an accuracy of 0.904, 0.841, 0.740, and 0.748 against MDPI, RUSA, RUSA-19, and UCL datasets, respectively. The results show that the SVM classifier and the Word2Vec CBOW (Continuous Bag of Words) model are more beneficial options for Roman Urdu sentiment analysis, but that BERT word embedding, two-layer LSTM, and SVM as a classifier function are more suitable options for English language sentiment analysis. The suggested model outperforms existing well-known advanced models on relevant corpora, improving the accuracy by up to 5%.

Keywords: sentiment analysis; Roman Urdu language; LSTM; machine learning; deep learning; word embedding

1. Introduction

As a result of the rise of low-cost mobile devices and ultra-fast Internet connection, users have recently been inspired to submit a variety of data on social networking sites such as Twitter, Facebook, and YouTube. User input on a variety of things, as well as on people's thoughts regarding services, online learning, and other issues, is included in these data [1,2]. As the use of social networking platforms expands, users are encouraged to share their opinions and emotions, and to participate in a variety of discussion groups [3–5]. To be more explicit, sentiment analysis (SA) is critical for comprehending people's actions [6–8].

The importance of SA may be seen in our need to understand how others respond to a situation and what they think [9]. Most businesses and governments are interested in obtaining important information from user comments, such as the emotions and feelings that underpin client opinions [10–13]. Natural language processing (NLP), data mining, text analysis, machine learning, and deep learning approaches are used to analyze the

feelings behind user-shared comments in SA [14,15]. Organizations and enterprises are interested in developing successful public relations strategies, running campaigns, overcoming weaknesses, and gaining more clients, as interest in SA grows. Businesses are eager to hear what customers have to say about their services and products [16,17]. Furthermore, political parties are interested in learning about their popularity among the public and what the media has to say about them.

In recent times, SA's focus has shifted to analyzing the emotions expressed in social media evaluations. The use of SA has expanded across a variety of spheres including harassment, politics, entertainment, sports, and medicine [18]. SA includes improved NLP approaches, data mining for predictive analysis, and contextual understanding of texts, all of which are current research issues [19,20].

Machine learning approaches such as support vector machine (SVM), logistic regression (LR) and naive Bayes (NB) have been used to solve diverse NLP tasks for many years. In several NLP-related applications, neural network (NN) methods created on dense vector representations have recently demonstrated state-of-the-art performances [21,22]. Deep learning NNs initially demonstrated impressive performance in computer vision and pattern recognition workloads. Many deep learning algorithms have been employed to handle complex NLP problems such as sentiment analysis as a result of this trend.

SA has received a great deal of research attention from academics. Although English and European languages have received considerable academic attention because they are considered rich dialects in terms of the tools and procedures needed to conduct SA, there are a number of other dialects, such as Roman Urdu/Hindi, that are considered to be resource deprived [23]. Urdu is the native and official language of Pakistan, and it is extensively spoken in many Indian states and Union territories. Roman Urdu and Romanaagari are names for the Urdu and Hindi languages written in the English alphabet, respectively. The majority of people in Pakistan, India, and other South Asian nations use Romanaagari or Roman Urdu script to interact on social media networks.

However, compared to other languages such as English, very few research investigations have concentrated on Roman Urdu/Hindi, due to resource constraints such as a lack of lexical resources and morphological issues. The challenge of Roman Urdu/Hindi sentiment analysis has not yet been fully investigated, despite the widespread use of these languages. Hence, the main subject of this research study is Roman Urdu sentiment analysis.

The primary contribution of this research is to introduce a novel deep learning method designed on CNN-LSTM for Roman Urdu and English sentiment analysis based on user-generated textual data on social media, using various word embedding models, and to compare the performances of Word2Vec (CBOW and skip-gram), GloVe, FastText, and TF-IDF words-to-vectors models on Roman Urdu text classification.

The remainder of this study is divided into five sections. Section 2 discusses related work on Roman Urdu sentiment analysis. Section 3 outlines our recommended methodology. The experimental setup is described in Section 4 of this paper. The experimental results and assessments of the proposed methodology are explained in Section 5. The final section brings the study to a conclusion and discusses future research.

2. Literature Review

Various scholars have expended a great deal of effort to construct models, datasets, and other technological resources for Roman Urdu/Hindi sentiment analysis. However, the difficulties and challenges of Roman Urdu/Hindi sentiment analysis have not yet been thoroughly examined.

Many websites were crawled to construct a Roman Urdu corpus, which contains sentiments regarding various items and services, according to [24]. To classify the textual data, the authors used various machine learning methods such as NB, LR, and SVM. The study's findings showed that SVM outperformed other machine learning classifiers. In another investigation [25], the researchers gathered 300 negative and positive sentiments

from a blog in English and Roman Urdu. They then performed a sentiment analysis using three distinct machine learning models: KNN, NB, and decision tree. They discovered that NB outperformed the other two classifiers in terms of accuracy. Khan and Malik produced a corpus of user reviews in [26] by scraping several vehicle websites and categorizing them as negative or positive. The reviews were classified using random forest, multi-nominal NB, decision tree, KNN, and SVM machine learning classifiers. The experimental results showed that multi-nominal NB had the highest accuracy of all the algorithms examined. In a recent study [27], Mehmood et al. created a Roman Urdu corpus with only 779 user reviews from five different genres, including movies, politics, mobile phones, miscellaneous, and theater. The researchers combined n-gram features with five machine learning models: LR, NB, KNN, SVM, and DT. Compared to the other machine learning algorithms, NB produced better outcomes (Table 1).

Table 1. Summary and comparison of existing literature.

Reference	Used Corpus	Classification Algorithms	Learning Method	Embedding and Features	Accuracy (%)
[28]	15,000 user reviews gathered from various websites	Neural Network	Supervised	-	80
[29]	Only 300 reviews of positive and negative classes	NB, DT, KNN	Supervised	-	71
[30]	3241 sentences of positive, negative, and neutral classes	SVM, LR, NB, RCNN, Hybrid Multichannel Approach	Supervised	Word2Vec, GloVe and FastText	82
[31]	10,021 user sentences from various websites of three classes	RCNN, Rule-based and N-gram	Supervised and Unsupervised	Word2Vec	75
[32]	11,000 reviews of two classes	KNN, DT, RF, LR, NB, ANN, SVM, AdaBoost, wVoting	Supervised	-	82
[33]	11,000 sentences of two classes	KNN, DT, RF, LR, NB, ANN, SVM, AdaBoost, wVoting	Supervised	-	81

Another study on Roman Urdu sentiment analysis [34] was conducted. The researchers used a hybrid model to examine Roman Urdu sentiment analysis using several lexicons and machine learning technologies. To categorize text data, they employed SVM and NB classifiers. Similarly, researchers presented a new feature representation approach for sentiment analysis in another study [35]. The performance of the suggested feature representation, dubbed "Discriminative Feature Spamming", was compared to binary weighting TF and TF-IDF, with character- and word-level n-gram features utilizing NB, LR, weighted voting, majority voting, and multi-layer perceptron models (MLP). The suggested feature representation method greatly improved the performance of all machine learning algorithms, according to the results. Mehmood et al. [32] constructed a Roman Urdu dataset with reviews from six different genres in another research study. To address the extensive morphological structure of the Roman Urdu dialect, they suggested a methodology that utilized character-level, word-level, and a mix of word- and character-level features. They were able to improve the performance of machine learning classifiers by 12% above the baseline by doing so. On the other hand, deep learning classifiers such as recurrent neural network (RNN) and LSTM have shown promising results in a variety of NLP tasks. Another important study [31] tackled Roman Urdu sentiment analysis using a recurrent convolutional neural network (RCNN). The study's main contribution was to provide a state-of-the-art manually labeled corpus for the resource-scarce Roman Urdu language. The authors used a variety of models to categorize user evaluations, including rule-based, n-gram, and RCNN. There were two types of experiments: tertiary (neutral, negative, and positive) classifications and binary (negative and positive) classifications. The results revealed that RCNN was able to outperform the other tested approaches.

The most important work [36] has recently been done by employing various machine learning and deep learning techniques for SA of Urdu text. Firstly, user reviews in Urdu from six different domains were gathered from various social media platforms to create a state-of-the-art corpus. Human specialists later carefully annotated the entire Urdu corpus. Finally, the created Urdu corpus was validated using a combination of machine learning techniques such as RF, NB, SVM, AdaBoost, MLP, and LR, and deep learning algorithms such as LSTM and CNN-1D. LR algorithms outperformed all other machine learning and deep learning algorithms in terms of accuracy.

In [30], another attempt was made to solve the problem of Roman Urdu sentiment analysis using supervised machine learning and deep learning models with word embedding. The authors gathered 3241 Roman Urdu positive, negative, and neutral feelings. SVM, LR, and NB were among the machine learning algorithms used. The authors also evaluated their corpus using a hybrid multi-channel method, which included testing deep learning techniques such as RCNN and RNN. They used three neural word embedding approaches in their suggested hybrid approach: Word2Vec, GloVe, and FastText. In terms of accuracy, F1 score, precision, and recall, their suggested hybrid multi-channel framework outperformed the other applied machine and deep learning methods by a wide margin.

Similarly, LSTM with a word embedding layer was used to test another [27] deeplearning-based model for Roman Urdu sentiment analysis. The input layer, word embedding layer, LSTM layer, and final output layer were the four layers of the suggested methodology. The proposed model outperformed the competition in terms of accuracy. Table 1 represents a short summary of the literature. Some examples of to neutral, positive, and negative classes of Roman Urdu/Hindi text are presented in Figure 1.

Positive review examples	Negative review examples	Neutral review examples
Eik Oumda Software hai (An excellent software)	Ap ka kaam Baoht Kharab hai (Your work is very bad)	G haan speed pehlay se kam hai laikin chaal to raha hai (Yes, its speed is slower but at least it's working properly)
Ap ka Kaam lajawab (Your work is awesome)	Ye to bahot porana ho chuka hai. (It's too old. . . !)	Achi hai magar porani ho gai hai, (Its good but old)
Ye Aik Bohat he umdah nuskah ha. (It is a very good recipe)	Bakwas (Rubbish)	jeetna aur harna kheil ka aik hisa, (Winning and losing is part of the game)

Figure 1. Examples of Roman Urdu/Hindi user reviews with English translations.

In [37], the authors analyze several categorical ways to leverage normative databases as a way of processing text with a dimensional model for the categorical models. Three dimensionality reduction strategies were evaluated: latent semantic analysis (LSA), probabilistic latent semantic analysis (PLSA), and non-negative matrix factorization (NMF). A normative database was used to create three-dimensional vectors (valence, arousal, dominance). The results revealed that dimensional modeling and the NMF categorical model performed the best. In another study [38] EmoBank discriminates between writer and reader emotions, whereas a subset of the corpus uses categorical VAD annotations based on basic emotions. In [39], the authors use Twitter data to understand the method for conveying the message in a different language, and they also analyze systems for bias towards specific races or genders. The authors proposed a neural-network-based approach in [40] for multidimensional emotion regression, which gives the rate of multiple emotion

dimensions for an input text automatically. A discriminator was improved with adversarial training between two attention layers. In [41], the authors propose a tree-structured regional CNN-LSTM model to predict VA (valence–arousal) ratings in texts, while in [42] the authors present a multidimensional relation model to predict the dimension scores in deep neural networks.

Another trend is to use sentiment embeddings by injecting sentiment knowledge into traditional word embeddings. Learning sentiment-specific word embeddings, called sentiment embeddings, is proposed in [43]. In sentiment embeddings, authors encode the sentiment information of texts along with the context information of words. Similarly, in [44], the authors provide an embedded word learning architecture that utilizes local context information as well as global sentiment representation. Their architecture is applicable to sentences and also at the document level. In [45], the authors present a model for improving existing pre-trained word vectors by using real-valued sentiment intensity scores derived from sentiment lexicons instead of creating a new embedding from a labeled corpus.

Scholars have recently become more interested in attention-based approaches. By giving various weights to different sections of the context, the attention technique is utilized to stress the relevant aspects of the context. For sentiment analysis, Basiri et al. [46] used a CNN bidirectional LSTM and the GRU attention mechanism. To place less or more focus on particular words, the attention module was employed on the outputs of bidirectional layers of LSTM and GRU. Several tests were carried out on five separate datasets. The proposed model outperformed other existing models, according to the findings. In a similar study [47], the authors utilized CNN with max pooling for feature extraction and bi-LSTM for capturing long-term dependencies. Finally, the authors used an attention mechanism to place focus on individual words. Four different datasets were utilized to train the model. Their model outperformed some of the baseline outcomes. Similarly, the authors in [48] developed an attention technique to handle sentiment classification at the aspect level. Five separate benchmark datasets were used to validate their proposed model. The obtained findings demonstrated the model's efficacy.

3. Methodology of Research

In this paper, we implement deep Roman Urdu SA using CNN-LSTM, a deep learning model for Roman Urdu/Hindi sentiment analysis that combines word embedding methods such as Word2Vec, GloVe, TF-IDF, BERT and FastText to a convolutional neural network(CNN) architecture. CNN cannot be utilized to generate long-distance dependencies from input text data due to the locality of the pooling and convolutional layers; however, a recurrent neural layer can effectively overcome this problem [6]. Therefore, the long-distance dependencies are captured using LSTM in our suggested model. Finally, the fully connected layer is passed on to machine learning classifiers such as SVM, NB, DT, RF, KNN, and softmax to classify the reviews into different categories (negative, positive, or neutral).

To the best of our knowledge, this is the first time a deep CNN-LSTM hybrid model has been proposed for Roman Urdu text classification. The suggested model's priority overhead is that CNN is utilized for feature selection and LSTM layers are employed to capture long-term text data dependencies. In addition, the proposed model includes an additional LSTM layer, which improves the performance. The suggested model is also equipped with the most up-to-date word embedding techniques, such as BERT. Other current models employed the softmax function for classification, but we used typical machine learning classifiers. In a nutshell, our model includes the most recent word embedding, machine learning, deep learning, CNN, and recurrent neural network approaches. Figure 2 depicts the information flow and basic architecture, whereas Figures 3 and 4 depict the CNN and LSTM basic architectures, respectively.
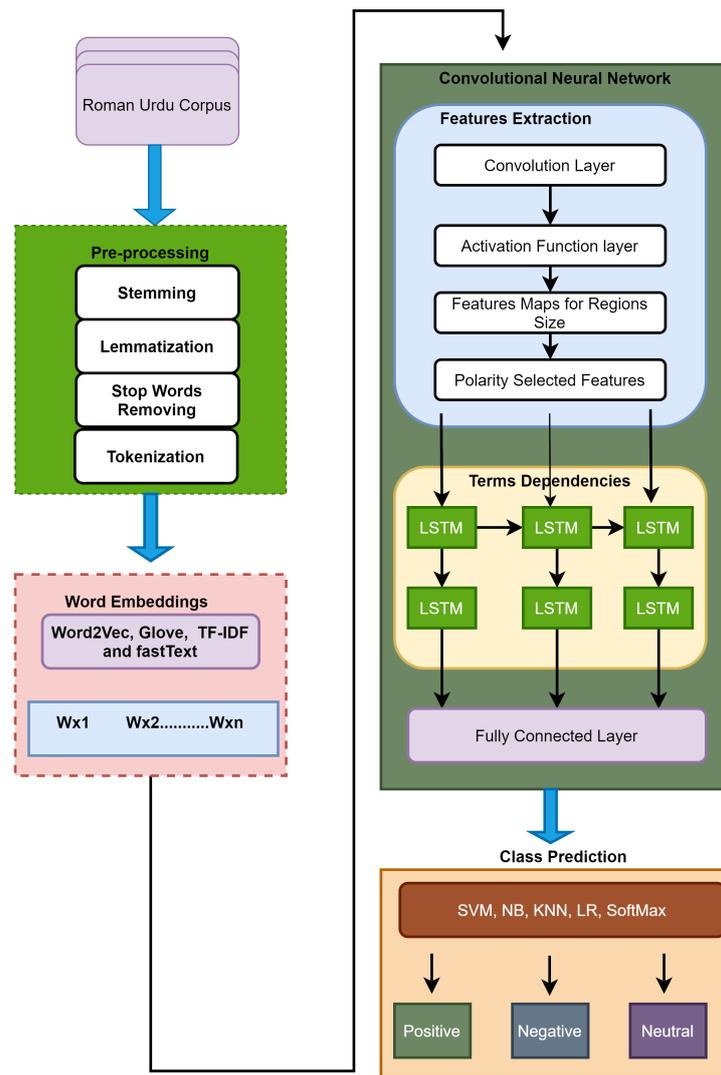
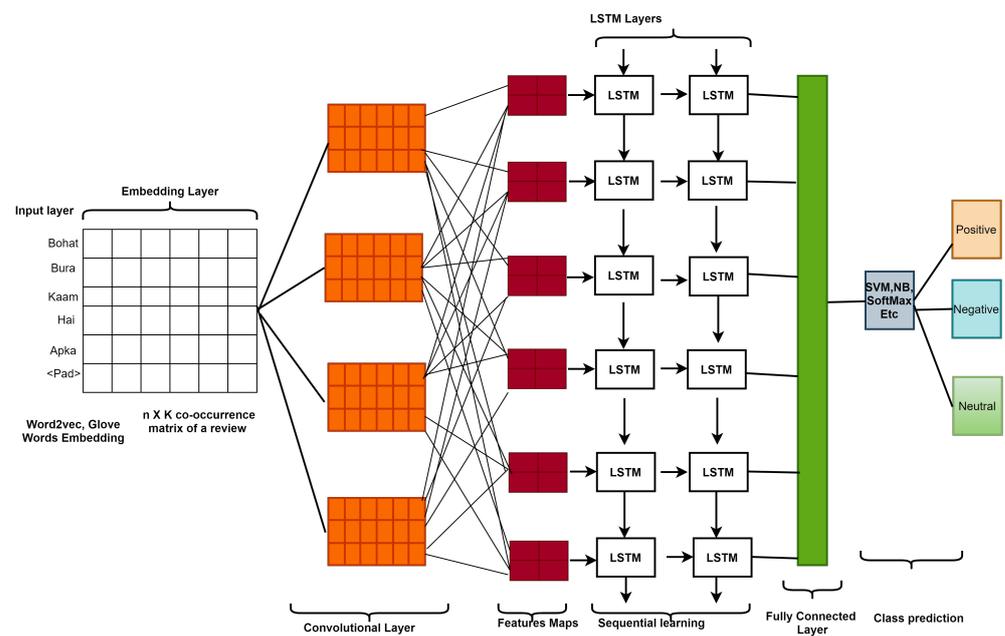**Figure 2.** Abstract-level architecture of proposed CNN-LSTM Roman Urdu sentiment analysis.

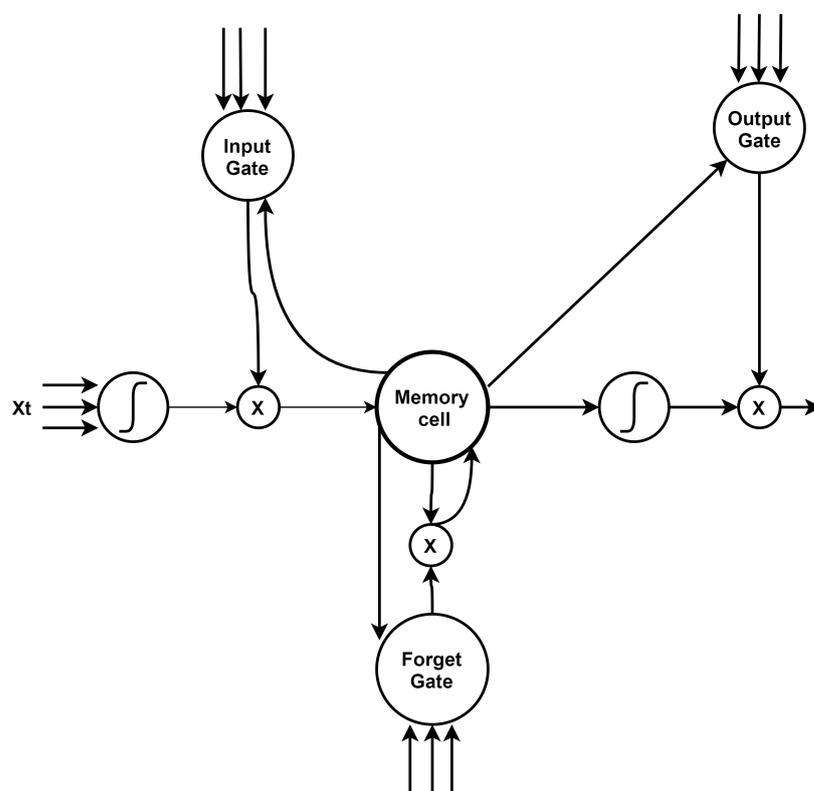**Figure 3.** CNN-LSTM framework of Roman Urdu sentiment analysis.

Figure 4. LSTM architecture with memory cell.

3.1. Word Embedding

Word2Vec [49] is a neural word-to-vector model that uses surrounding words to predict the vector of a required word. The skip-gram and Continuous Bag of Words (CBOW) learning techniques are the two most common learning strategies used in the Word2Vec embedding model. In comparison to distant words, the skip-gram technique of prediction gives more weight to adjacent or close words, whereas the CBOW approach does not affect the sequence of neighboring words because it predicts on the basis of the existing word to close the gap between context words. Both CBOW and skip-gram learn combined vector representations for each word, using only local context.

Unlike Word2Vec, GloVe [50] neural word embedding considers the entire context of words. A neural network is used in GloVe word embedding to break down a co-occurring matrix into a word vector. GloVe embedding [50] outperforms Word2Vec [50] in word similarity and analogy tests because the GloVe embedding model considers the association between word pairs and adds supplementary meanings to the neural network. GloVe embedding also reduces the weights of frequent word pairs such as those including "the", "a", and so on. The GloVe model, however, is built on a co-occurrence matrix, which necessitates a large amount of memory for storage.

FastText [51], like Word2Vec, learns the vector representation of each word as well as the n-grams placed within each word, in order to properly learn the representation of out-of-vocabulary (OOV) words, which is a common challenge encountered by both Word2Vec and GloVe. Subsequently, at each training step, the representation values are averaged to generate a single vector. Although these embedding models are more computationally expensive than Word2Vec and GloVe, they allow neural word embedding to encode significant sub-word information. Compared to Word2Vec, FastText neural word embedding models are significantly more accurate.

BERT [52] is an acronym for bidirectional encoder representations from transformers. BERT aims to condition both right and left context across all levels to pre-train deep bidirectional representations from unlabeled data. As a result, the pre-trained BERT model

may be fine-tuned with only one new output layer, to provide state-of-the-art models for a variety of tasks such as sentiment analysis, question answering, and language translations, without requiring significant task-specific design changes. BERT is empirically powerful and abstractly simple. On eleven various NLP-related tasks, it delivered new state-of-the-art results.

3.2. CNN-LSTM

Suppose $X_i \in R^k$ represents the K-dimensional vector which is equal to the *ith* token in a user review with total size or length of *n*, which is denoted as a string of its word vectors, mathematically shown in Equation (1). If the length of the sentence is less than *n*, then zero padding is applied.

$$X_{1:n} = X_1 + X_2 + X_3 \ldots\ldots\ldots + X_n \tag{1}$$

The $+$ operator denotes a concatenation operation in Equation (1). Similarly, suppose the concatenation of the words X_i, X_{i+1}, X_{i+2}, $\ldots\ldots X_{i+j}$ is equivalent to $X_{i:i+j}$. Let $W \in R^{hk}$ denote the convolutional filters, applied in an *nXK*-dimensional matrix of a sentence with a window or gap of h words, to produce a new feature matrix. The basic element $X_{i:i+j}$ denotes the local feature matrix from the *ith* to the $(I+J)th$ line of the present sentence vector. Equation (2) can be used to produce a feature C_{if} from a window of words $X_{i:i+h-1}$.

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

where *b* denotes bias and belongs to the set of real numbers, and *f* is an activation function such as hyperbolic tangent and sigmoid. The convolutional filter is convoluted on each window of words to generate a feature map by applying Equation (3).

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

where *C* belongs to R^{n-h+1}.

The above is the mathematical technique for constructing a single feature map from a single convolutional filter. A convolutional layer with multiple *m* filters will produce $m(n-h+1)$ features in the same way. Because feature selection can disrupt long-term dependencies early in the LSTM layers, the max pooling layer is not employed on features maps. The features are directly transferred into the LSTM layer before the fully linked layer, to capture long-term dependencies.

LSTM can capture long-term relationships in sentences of unknown length and can be effectively utilized to govern information by avoiding a vanishing gradient. The basic architecture of the LSTM model is depicted in Figure 4. In an LSTM, the memory cell is utilized to save the selected data for longer without decaying.

To execute the current input data, LSTM used recursive execution of the present memory cell using the current input X_t and the previous hidden state h_{t-1}, where *t* denotes the present time and $t-1$ denotes the previous time. Additionally, LSTM has an input gate i_t, output gates O_t, and forget gates f_t, where C_t^- represents the present memory cell at time *t*. The following mathematical equations explain the operational structure of LSTM. Equations (4) and (5) are used to calculate the values of the input gate and current memory cell states at time *t*.

$$i_t = \sigma(W_i \cdot X_t + U_i \cdot h_{t-1} + b_i) \tag{4}$$

$$C^- = \tanh(W_c \cdot X_t + U_c \cdot h_{t-1} + b_c) \tag{5}$$

The value of the forget gate at time *t* is calculated using Equation (6).

$$f_t = \sigma(W_f \cdot X_t + U_f \cdot h_{(t-1)} + b_f) \tag{6}$$

Similarly, Equation (7) is used to calculate the value of the new state of the memory cell at time t.

$$C_t = i_t \cdot C^- + f_t + C_{t-} \tag{7}$$

The values of the output gate are calculated using Equations (8) and (9).

$$O_t = \sigma(W_o \cdot X_t + U_o \cdot h_{(}t-1) + b_o + V_o \cdot C_t) \tag{8}$$

$$h_t = O_t \cdot \tanh C_t \tag{9}$$

The input of the memory cell is denoted by X_t at time t, where W_f, W_i, W_c, W_o, V_o, U_o, U_i, U_c, and U_f all are weight matrices, σ represents a sigmoid function and b_c, b_f, b_i, and b_o are bias vectors. Throughout training, our supposed model learns the values of U_i and W_i. The values of the forget gate, input gate, and output gate are in the range [0,1]. In this proposed model, after feature mapping, the output or results of the first LSTM layer are fed to the second LSTM layer, which generates the deep representation of the input user review. The ultimate result or outputs of the LSTM are fused into a matrix. Finally, this matrix is fed to the CNN's fully connected layer.

The basic CNN-LSTM design has been widely employed in previous studies [53–55], However, the CNN-LSTM model we propose is unique for the following reasons: (1) in comparison to previous studies, our suggested model includes one additional layer of LSTM to improve the performance; (2) while the majority of previous studies used softmax as a classification function, we included traditional machine learning models such as NB, DT, KNN, RF, and SVM in our suggested model. The proposed and existing models further differ in that we study the CNN-LSTM architecture using a variety of word embedding models including fixed (GloVe, Word2Vec), pre-trained (FastText), and context-based (BERT) models, whereas previous studies solely employed fixed word embedding models.

3.3. Experimental Datasets

The experimental details and configuration of our proposed CNN-LSTM Roman Urdu sentiment analysis model on several datasets are described in this section. The RUSA-19 [31], UCL Roman Urdu corpus [28], RUSA dataset [32,33] and IMDb movie reviews created in [56] were used to test our proposed CNN-LSTM Roman Urdu sentiment analysis model.

The RUSA-19 dataset contains 10,021 Roman Urdu user reviews that cover a wide range of topics including movies, drama, food, electronics, software, blogs, and sports. These comments were gathered from various social media channels. In the RUSA-19 corpus, all of these reviews are classified as positive (represented by 1), negative (represented by 2), or neutral (represented by 0). There are 3778 positive reviews, 2941 negative reviews, and 3302 neutral reviews in the RUSA-19 corpus.

Similarly, the UCL Roman Urdu corpus contains 20,228 sentences that are divided into three categories: neutral, positive, and negative. There are 5286 negative sentences, 6013 positive sentences, and 8929 neutral sentences in the UCL Roman Urdu dataset.

Our suggested model is validated using another Roman Urdu RUSA dataset [33]. RUSA contains 11,000 Roman Urdu evaluations from six different genres including music, food, mobile, movie/drama, sports, and politics. These reviews were gathered from a variety of social media sites, including youtube.com, facebook.com, and twitter.com. In the RUSA dataset, there are 5314 reviews in the negative category and 5686 sentences in the positive category. Three native speakers manually annotated the data.

The proposed model is also tested using an English language corpus called IMDB movie reviews, which was created in [56] and contains 50k negative and positive movie reviews. The authors have already divided the IMDB dataset into equal testing and training sections, with each section including 12.5k positive and negative ratings.

During the experiments, many settings, scenarios, and hyper-parameters were tested and tried. For the CNN experimental parameters, the applied convolutional layer employed different size fitters such as 3, 4, and 5 with 256-feature maps. The ReLU function is used as an activation function. To reduce the problem of overfitting, the recurrent layer's

dropout was set to 0.5 earlier. The LSTM uses a sigmoid function as an activation function, with 128 hidden states. A total of 50 epochs are used throughout the experiment. The experiments were written in Python using the TensorFlow framework.

4. Results and Discussion

The proposed CNN-LSTM model for Roman Urdu sentiment analysis is trained on various datasets, as shown in Table 2, then the performance of the proposed model is assessed using various evaluations such as accuracy, precision, recall, and F1-score. Each dataset is divided into testing and training sets. For training, 80% of the data were used, and 20% were used for testing, for each corpus.

Table 2. Statistics of used datasets.

Dataset	Positive Reviews	Negative Reviews	Neutral Reviews	Total Reviews
RUSA-19	3778	2941	3302	10,021
UCL	6013	5286	8929	20,228
RUSA dataset	5686	5314	-	11,000
IMDb movie reviews	25,000	25,000	-	50,000

The excellence of the word-to-vector models and feature extraction methods is ultimately determined by the model performance. Therefore, employing NB KNN, RF, LR, softmax, and SVM classifiers as classification functions after the fully connected layers, we comprehensively evaluated the results of the proposed CNN–LSTM Roman Urdu sentiment analysis. The SVM model outperformed all other used classifiers in terms of accuracy, precision, recall, and F1-score, which is consistent with [32]. Tables 3–5 present the achieved results of our proposed model on UCL, RUSA-19, and the RUSA corpus, respectively. The proposed CNN-LSTM model achieved accuracies of 0.740, 0.748, 0.841, and 0.904 on UCL, RUSA-19, RUSA and IMDB datasets, respectively. On the other hand, the CNN-LSTM performance remained slightly low with the softmax function. Similarly, the performance of proposed model with KNN (K = 10) and NB as classifier functions remained comparatively better than with the softmax function.

Table 3. Roman Urdu CNN-LSTM performance with various ML classifiers using UCL corpus.

ML Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)
Random Forest	0.712	0.725	0.690	0.707
Logistic Regression	0.706	0.700	0.695	0.697
SVM	0.740	0.750	0.714	0.731
NB	0.716	0.731	0.690	0.709
KNN	0.710	0.714	0.670	0.691
Softmax	0.695	0.690	0.673	0.681

LSTM, however, is a deep feed-forward neural network model. It has been established that the number of LSTM layers has an effect on classification performance. In the supposed CNN–LSTM Roman Urdu sentiment analysis model, we observed the performance using two LSTM layers vs. one LSTM layer, with 128 units in each LSTM layer. Table 6 summarizes the findings of this investigation. Two stacked LSTM layers increased classification outcomes by +2.10% in accuracy, 1.80% in precision, and 0.90% in recall, compared to a single-layer LSTM. Similarly, our proposed model achieved slightly better results with two-layer LSTM for all used datasets. As a result, two LSTM layers are considered adequate for constructing higher-order feature representations of Roman Urdu phrases so that they may be more easily classified. The proposed model achieved accuracies of 0.719, 0.740,

and 0.839 with one-layer LSTM against UCL, RUSA-19, and RUSA datasets, respectively. On the other hand, slightly better accuracies of 0.740, 0.748, and 0.841 were achieved by applying two-layer LSTM on UCL, RUSA-19, and RUSA datasets, respectively.

Table 4. Roman Urdu CNN-LSTM performance with various ML classifiers using RUSA-19 corpus.

ML Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)
Random Forest	0.695	0.693	0.673	0.682
Logistic Regression	0.698	0.701	0.674	0.687
SVM	0.748	0.762	0.729	0.745
NB	0.729	0.735	0.694	0.713
KNN	0.708	0.719	0.675	0.696
Softmax	0.670	0.663	0.648	0.655

Table 5. Roman Urdu CNN-LSTM performance with various ML classifiers using RUSA dataset.

ML Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)
Random Forest	0.770	0.770	0.762	0.765
Logistic Regression	0.760	0.762	0.758	0.759
SVM	0.841	0.850	0.840	0.844
NB	0.811	0.812	0.800	0.805
KNN	0.789	0.792	0.783	0.787
Softmax	0.755	0.753	0.745	0.748

Table 6. Effect of LSTM layers on proposed Roman Urdu CNN-LSTM model.

Dataset	No of LSTM Layers	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)
UCL	1	0.719	0.732	0.705	0.718
	2	0.740	0.750	0.714	0.731
RUSA-19	1	0.740	0.744	0.706	0.724
	2	0.748	0.762	0.729	0.745
RUSA dataset	1	0.839	0.830	0.821	0.825
	2	0.841	0.850	0.840	0.844

Various fixed and pre-trained word embedding techniques such as Word2Vec, GloVe, TF-IDF, and FastText, were used to test the classification results of the proposed CNN–LSTM Roman Urdu SA. Table 7 compares the accuracy of Word2Vec (skip-gram and CBOW), GloVe, TF-IDF, and FastText CNN–LSTM models. According to the findings, Word2Vec-based approaches (CBOW and skip-gram) outperformed FastText, GloVe, and TF-IDF with accuracies of 0.841 and 0.837 for the RUSA dataset. Similarly, Word2Vec-based models (CBOW and skip-gram) achieved the highest accuracy and F1-score for other corpuses used in this study.

Table 7. CNN-LSTM Roman Urdu model accuracy using various word embeddings with two-layer LSTM.

Word Embedding	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)
Word2Vec (CBOW)	UCL	0.740	0.750	0.714	0.731
	RUSA-19	0.748	0.762	0.729	0.745
	RUSA dataset	0.841	0.850	0.840	0.844
Word2Vec (skip-gram)	UCL	0.735	0.741	0.708	0.724
	RUSA-19	0.739	0.752	0.720	0.735
	RUSA dataset	0.837	0.835	0.824	0.829
TF-IDF	UCL	0.714	0.713	0.691	0.701
	RUSA-19	0.726	0.722	0.700	0.710
	RUSA dataset	0.815	0.801	0.791	0.795
GloVe	UCL	0.728	0.728	0.711	o.719
	RUSA-19	0.736	0.738	0.718	0.727
	RUSA dataset	0.826	0.824	0.810	0.816
FastText	UCL	0.738	0.737	0.730	0.733
	RUSA-19	0.736	0.745	0.721	0.732
	RUSA dataset	0.828	0.823	0.816	0.819

The suggested model's findings against the IMDb movie reviews corpus are shown in the Table 8. Movies that received ratings of 4 or less from viewers were labeled as bad reviews, while those that received ratings of 7 or more were labeled as positive reviews. The reviews of movies within a rating of 7 to 4 were excluded from the corpus. Across a variety of datasets and languages, the suggested model exhibited the same pattern of performance. With two-layer LSTM, our proposed CNN-LSTM model with BERT as an embedding layer obtained the maximum accuracy of 0.904. The BERT embedding model outperforms the others due to the transformer and self-attention technologies. Due to the fixed word-to-vector representations, the suggested model achieved less accuracy with fixed word embedding approaches such as Word2Vec. Pre-trained word embedding, such as FastText, on the other hand, yielded better outcomes than Word2Vec, since FastText is pre-trained on the English language. Similarly, the additional LSTM second layers improved the performance of the proposed model. As a classifier function, the SVM model surpassed all other models such as NB, KNN, RF, and LR.

We conducted various experiments on various datasets to confirm the results of the CNN–LSTM Roman Urdu SA against existing models. The UCL dataset includes 20,228 user reviews gathered from a variety of social media sources. The RUSA-19 dataset contains 10,021 Roman Urdu user reviews. The RUSA dataset includes 11,000 negative and positive Roman Urdu user reviews gathered from various social media platforms such as YouTube, Facebook, etc. The CNN–LSTM Roman Urdu SA classification accuracy was compared to research study [31], recurrent convolutional neural network and rule-based approaches were used. The study in [32] used various machine learning algorithms such as RF, SVM, DT, NB, KNN, LR, ANN, AdaBoost and wVoting. The research study in [33] implemented various machine learning algorithms such as RF, SVM, DT, NB, KNN, LR, ANN, AdaBoost and wVoting with word-gram and character-gram features union. Table 9 compares the achieved results of the CNN–LSTM Roman Urdu sentiment analysis with the respective corpuses to the performance of the other existing techniques, in terms of accuracy, precision,

recall, and F1-score. The proposed CNN–LSTM Roman Urdu sentiment analysis attained the highest performance in all used corpuses with 0.740, 0.748, 0.841, and 0.750 for the accuracy, precision, recall, and F1-score, respectively, for the UCL dataset. Similarly, the proposed model achieved 0.762, 0.850, 0.714, and 0.729 for the accuracy, precision, recall, and F1-score, respectively, for the RUSA-19 dataset and 0.840, 0.731, 0.745, and 0.844 for accuracy, precision, recall, and F1 score, respectively, for the RUSA dataset. This assessment shows the effectiveness of the suggested CNN-LSTM deep learning technique for Roman Urdu SA.

Table 8. CNN-LSTM model accuracy using various word embeddings with two-layer LSTM using IMDb movie reviews corpus.

Word Embedding	Classifier	LSTM Layers	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)
Word2Vec	Random Forest	1	0.847	0.840	0.850	0.844
		2	0.853	0.847	0.854	0.850
	Logistic Regression	1	0.851	0.845	0.855	0.849
		2	0.852	0.848	0.856	0.851
	SVM	1	0.854	0.848	0.859	0.853
		2	0.856	0.851	0.861	0.855
	NB	1	0.851	0.847	0.857	0.850
		2	0.855	0.848	0.860	0.853
	KNN	1	0.848	0.843	0.851	0.846
		2	0.850	0.845	0.853	0.848
FastText	Random Forest	1	0.862	0.859	0.860	0.859
		2	0.866	0.863	0.865	0.863
	Logistic Regression	1	0.865	0.862	0.866	0.863
		2	0.868	0.865	0.868	0.866
	SVM	1	0.871	0.869	0.871	0.869
		2	0.873	0.872	0.874	0.872
	NB	1	0.856	0.849	0.862	0.855
		2	0.861	0.853	0.866	0.859
	KNN	1	0.851	0.847	0.855	0.850
		2	0.856	0.849	0.858	0.853
BERT	Random Forest	1	0.889	0.885	0.890	0.887
		2	0.894	0.888	0.894	0.890
	Logistic Regression	1	0.892	0.887	0.892	0.889
		2	0.895	0.890	0.895	0.892
	SVM	1	0.895	0.890	0.896	0.892
		2	**0.904**	**0.895**	**0.903**	**0.898**
	NB	1	0.892	0.887	0.892	0.889
		2	0.896	0.889	0.895	0.891
	KNN	1	0.889	0.883	0.890	0.886
		2	0.892	0.886	0.894	0.889

The confusion matrix is a measure for assessing the validity of a classification. Figures 5–7 show the proposed model's confusion matrix for the UCL, RUSA-19, and RUSA datasets, respectively. Only 10.60% of positive reviews were mislabeled as negative and 13.90% as neutral, while 75.50% of positive reviews were classified accurately as positive. Only 13.00% and 11.00% of negative reviews were misclassified as positive and neutral, respectively, while 76.00% of negative reviews were classified accurately

as negative. Only 15.40% and 14.10% of neutral reviews were misclassified as positive and negative, respectively, while 70.50% of negative reviews were classified accurately as neutral using CNN–LSTM Roman Urdu sentiment analysis on the UCL corpus.

Table 9. Proposed CNN-LSTM Roman Urdu model comparison with existing models.

Reference	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Measure (%)
[31]	UCL	0.693	0.732	0.699	0.715
	RUSA-19	0.713	0.710	0.683	0.696
[33]	RUSA dataset	0.811	0.816	0.828	0.822
[32]	RUSA dataset	0.821	-	-	-
Proposed model	UCL	**0.740**	**0.750**	**0.714**	**0.731**
	RUSA-19	**0.748**	**0.762**	**0.729**	**0.745**
	RUSA dataset	**0.841**	**0.850**	**0.840**	**0.844**

Predicted/Actuals	Positive	Negative	Neutral
Positive	75.50 %	10.60 %	13.90 %
Negative	13.00 %	76.00 %	11.00 %
Neutral	15.40 %	14.10 %	70.50 %

Figure 5. Confusion matrix of CNN-LSTM Roman Urdu sentiment analysis with UCL corpus.

Predicted/Actuals	Positive	Negative	Neutral
Positive	76.12 %	12.20 %	11.68 %
Negative	9.08 %	77.02 %	13.90 %
Neutral	14.88 %	13.90 %	71.22 %

Figure 6. Confusion matrix of CNN-LSTM Roman Urdu sentiment analysis with RUSA-19 corpus.

Predicted/Actuals	Positive	Negative
Positive	84.30%	15.70 %
Negative	16.10 %	83.90 %

Figure 7. Confusion matrix of CNN-LSTM Roman Urdu sentiment analysis with RUSA dataset.

Only 12.20% of positive reviews were mislabeled as negative and 11.68% as neutral, while 76.12% of positive reviews were classified accurately as positive. Only 9.08% and 13.90% of negative reviews were misclassified as positive and neutral, respectively, while 77.02% of negative reviews were classified accurately as negative. Only 14.88% and 13.90%

of neutral reviews were misclassified as positive and negative, respectively, while 71.22% of negative reviews were classified accurately as neutral using CNN–LSTM Roman Urdu sentiment analysis on the RUSA-19 corpus. The example "jeetna aur harna kheil ka aik hisa hai", which means "winning and losing are both part of the game", is a neutral review from the RUSA-19 corpus that was correctly classified as such by the suggested model using Word2Vec, two-layer LSTM, and SVM as classification functions, but the same review was classified as a negative review using TF-IDF, one-layer LSTM, and softmax as classification functions. Similarly, utilizing GloVe embedding, the negative review "ye to baoht porana ho chuka hai" translated as "it's too old..." was recognized as a neutral review by the proposed model. However, Word2Vec and FastText word embedding accurately classified the same review.

Only 15.70% of positive reviews were mislabeled as negative, while 84.30% of positive reviews were classified accurately as positive. Only 16.10% of negative reviews were wrongly classified as positive, while 83.90% of negative reviews were classified accurately as negative using CNN–LSTM Roman Urdu sentiment analysis on the RUSA dataset.

Although all the above-mentioned work has been performed, there has been a great deal of effort with various models built and deployed on sentiment analysis for resource-rich languages such as English, but research in resource-impoverished languages such as Roman Urdu is still at an early stage. Due to a lack of training data and a diverse and complex morphological structure, pre-trained algorithms are less effective in Roman Urdu. We hope that our research will motivate academics to further investigate the Roman Urdu language.

5. Conclusions

Due to the recent epidemic, social media platforms have seen an exponential increase in user-generated material, which includes a wealth of data for various applications. SA is the study of social information in order to determine the preferences of the general population. It is problematic to accomplish SA in Roman Urdu, despite careful consideration of semantic and syntactic rules, as well as the input sentence's terms dependencies. As a result, this research developed a hybrid machine and deep learning model for English and Roman Urdu SA, which expertly combined a one-layer CNN model with two LSTM layers. For the input layer, this model is supported by a variety of word vector models. Experiments on three Roman Urdu datasets and one English dataset revealed that this model performed exceptionally well in Roman Urdu, with accuracy, precision, recall, and F1-scores of 0.841, 0.850, 0.840, and 0.844, respectively, and it performed equally well in English, with accuracy, precision, recall, and F1-scores of 0.904, 0.895, 0.903, and 0.898, respectively. The impact of word-to-vector approaches on sentiment classification in both dialects was thoroughly examined in this study, and it was discovered that the Word2Vec and BERT models were more appropriate options for acquiring semantic and syntactic information for Roman Urdu and English, respectively. Additionally, SVM, LR, NB, KNN, and softmax classifiers were implemented to assess the results of the suggested model. With an accuracy gain of up to 5%, SVM was shown to be the top-performing classifier. Due to the effectiveness of CNN in feature extraction and maintaining the long-term dependencies of LSTM, the suggested model outclassed well-known techniques on a number of benchmarks, improving accuracy by up to 5%.

Future research should investigate the use of self-attention models with various word embedding approaches in deep learning architectures for user interest discovery and recommendations for enhancing the results.

Author Contributions: L.K, A.A. and H.-T.C. contributed conception, designed the study, and analyzed the results. L.K. wrote the software and executed the experiments. A.A, L.K., K.M.A. and H.-T.C. wrote the first draft of the manuscript. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets presented in this study can be found in the following links: IMDB dataset: https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews, accessed on 10 February 2022, UCL Dataset: https://archive.ics.uci.edu/ml/datasets/Roman+Urdu+Data+Set, accessed on 10 February 2022, and Rusa-19 dataset: https://github.com/slab-itu/rusa_19, accessed on 10 February 2022.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Browne, G.J.; Walden, E.A. Is There a Genetic Basis for Information Search Propensity? A Genotyping Experiment. *MIS Q.* **2020**, *44*, 747–770. [CrossRef]
2. Mateen, A.; Khalid, A.; Khan, L.; Majeed, S.; Akhtar, T. Vigorous algorithms to control urban vehicle traffic. In Proceedings of the 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, Japan, 26–29 June 2016; pp. 1–5.
3. Al-Dabet, S.; Tedmori, S.; Mohammad, A.S. Enhancing Arabic aspect-based sentiment analysis using deep learning models. *Comput. Speech Lang.* **2021**, *69*, 101224. [CrossRef]
4. Ashraf, M.A.; Nawab, R.M.A.; Nie, F. Author profiling on bi-lingual tweets. *J. Intell. Fuzzy Syst.* **2020**, *39*, 2379–2389. [CrossRef]
5. Amjad, A.; Khan, L.; Chang, H.T. Effect on speech emotion classification of a feature selection approach using a convolutional neural network. *PeerJ Comput. Sci.* **2021**, *7*, e766. [CrossRef] [PubMed]
6. Hassan, S.U.; Imran, M.; Iftikhar, T.; Safder, I.; Shabbir, M. Deep stylometry and lexical & syntactic features based author attribution on PLoS digital repository. In *International Conference on Asian Digital Libraries*; Springer: Cham, Switzerland, 2017; pp. 119–127.
7. Shardlow, M.; Batista-Navarro, R.; Thompson, P.; Nawaz, R.; McNaught, J.; Ananiadou, S. Identification of research hypotheses and new knowledge from scientific literature. *BMC Med. Inform. Decis. Mak.* **2018**, *18*, 1–13. [CrossRef]
8. Thompson, P.; Nawaz, R.; McNaught, J.; Ananiadou, S. Enriching news events with meta-knowledge information. *Lang. Resour. Eval.* **2017**, *51*, 409–438. [CrossRef]
9. Sailunaz, K.; Alhajj, R. Emotion and sentiment analysis from Twitter text. *J. Comput. Sci.* **2019**, *36*, 101003. [CrossRef]
10. Khan, Z.; Iltaf, N.; Afzal, H.; Abbas, H. DST-HRS: A topic driven hybrid recommender system based on deep semantics. *Comput. Commun.* **2020**, *156*, 183–191. [CrossRef]
11. Hassan, S.U.; Aljohani, N.R.; Tarar, U.I.; Safder, I.; Sarwar, R.; Alelyani, S.; Nawaz, R. Exploiting Tweet Sentiments in Altmetrics Large-Scale Data. *arXiv* **2020**, arXiv:2008.13023.
12. Qadir, H.; Khalid, O.; Khan, M.U.; Khan, A.U.R.; Nawaz, R. An optimal ride sharing recommendation framework for carpooling services. *IEEE Access* **2018**, *6*, 62296–62313. [CrossRef]
13. Amjad, A.; Khan, L.; Chang, H.T. Semi-Natural and Spontaneous Speech Recognition Using Deep Neural Networks with Hybrid Features Unification. *Processes* **2021**, *9*, 2286. [CrossRef]
14. Xing, F.Z.; Pallucchini, F.; Cambria, E. Cognitive-inspired domain adaptation of sentiment lexicons. *Inf. Process. Manag.* **2019**, *56*, 554–564. [CrossRef]
15. Zhang, B.; Xu, X.; Li, X.; Chen, X.; Ye, Y.; Wang, Z. Sentiment analysis through critic learning for optimizing convolutional neural networks with rules. *Neurocomputing* **2019**, *356*, 21–30. [CrossRef]
16. Luo, Z.; Huang, S.; Zhu, K.Q. Knowledge empowered prominent aspect extraction from product reviews. *Inf. Process. Manag.* **2019**, *56*, 408–423. [CrossRef]
17. Ashraf, M.; Khan, L.; Tahir, M.; Alghamdi, A.; Alqarni, M.; Sabbah, T.; Khan, M. A study on usability awareness in local IT industry. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 427–432. [CrossRef]
18. Araque, O.; Zhu, G.; Iglesias, C.A. A semantic similarity-based perspective of affect lexicons for sentiment analysis. *Knowl.-Based Syst.* **2019**, *165*, 346–359. [CrossRef]
19. Safder, I.; Hassan, S.U. Bibliometric-enhanced information retrieval: A novel deep feature engineering approach for algorithm searching from full-text publications. *Scientometrics* **2019**, *119*, 257–277. [CrossRef]
20. Yadav, A.; Vishwakarma, D.K. Sentiment analysis using deep learning architectures: A review. *Artif. Intell. Rev.* **2020**, *53*, 4335–4385. [CrossRef]
21. Haydar, M.S.; Al Helal, M.; Hossain, S.A. Sentiment extraction from bangla text: A character level supervised recurrent neural network approach. In Proceedings of the 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), Rajshahi, Bangladesh, 8–9 February 2018; pp. 1–4.
22. Sze, V.; Chen, Y.H.; Yang, T.J.; Emer, J.S. Efficient processing of deep neural networks. *Synth. Lect. Comput. Archit.* **2020**, *15*, 1–341. [CrossRef]

23. Al-Ayyoub, M.; Khamaiseh, A.A.; Jararweh, Y.; Al-Kabi, M.N. A comprehensive survey of arabic sentiment analysis. *Inf. Process. Manag.* **2019**, *56*, 320–342. [CrossRef]

24. Rafique, A.; Malik, M.K.; Nawaz, Z.; Bukhari, F.; Jalbani, A.H. Sentiment analysis for roman urdu. *Mehran Univ. Res. J. Eng. Technol.* **2019**, *38*, 463–470. [CrossRef]

25. Bilal, M.; Israr, H.; Shahid, M.; Khan, A. Sentiment classification of Roman-Urdu opinions using Naïve Bayesian, Decision Tree and KNN classification techniques. *J. King Saud-Univ.-Comput. Inf. Sci.* **2016**, *28*, 330–344. [CrossRef]

26. Nazir, M.K.; Ahmad, M.; Ahmad, H.; Qayum, M.A.; Shahid, M.; Habib, M.A. Sentiment Analysis of User Reviews about Hotel in Roman Urdu. In Proceedings of the 2020 14th International Conference on Open Source Systems and Technologies (ICOSST), Lahore, Pakistan, 16–17 December 2020; pp. 1–5.

27. Ghulam, H.; Zeng, F.; Li, W.; Xiao, Y. Deep learning-based sentiment analysis for roman urdu text. *Procedia Comput. Sci.* **2019**, *147*, 131–135. [CrossRef]

28. Sharf, Z.; Rahman, S.U. Performing natural language processing on roman urdu datasets. *Int. J. Comput. Sci. Netw. Secur.* **2018**, *18*, 141–148.

29. Javed, I.; Afzal, H. Creation of bi-lingual social network dataset using classifiers. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*; Springer: Cham, Switzerland, 2014; pp. 523–533.

30. Mehmood, F.; Ghani, M.U.; Ibrahim, M.A.; Shahzadi, R.; Mahmood, W.; Asim, M.N. A precisely xtreme-multi channel hybrid approach for roman urdu sentiment analysis. *IEEE Access* **2020**, *8*, 192740–192759. [CrossRef]

31. Mahmood, Z.; Safder, I.; Nawab, R.M.A.; Bukhari, F.; Nawaz, R.; Alfakeeh, A.S.; Aljohani, N.R.; Hassan, S.U. Deep sentiments in roman urdu text using recurrent convolutional neural network model. *Inf. Process. Manag.* **2020**, *57*, 102233. [CrossRef]

32. Mehmood, K.; Essam, D.; Shafi, K.; Malik, M.K. Sentiment analysis for a resource poor language—Roman Urdu. *ACM Trans. Asian-Low-Resour. Lang. Inf. Process. (TALLIP)* **2019**, *19*, 1–15. [CrossRef]

33. Mehmood, K.; Essam, D.; Shafi, K.; Malik, M.K. An unsupervised lexical normalization for Roman Hindi and Urdu sentiment analysis. *Inf. Process. Manag.* **2020**, *57*, 102368. [CrossRef]

34. Hasan, A.; Moin, S.; Karim, A.; Shamshirband, S. Machine learning-based sentiment analysis for twitter accounts. *Math. Comput. Appl.* **2018**, *23*, 11. [CrossRef]

35. Mehmood, K.; Essam, D.; Shafi, K.; Malik, M.K. Discriminative feature spamming technique for roman urdu sentiment analysis. *IEEE Access* **2019**, *7*, 47991–48002. [CrossRef]

36. Khan, L.; Amjad, A.; Ashraf, N.; Chang, H.T.; Gelbukh, A. Urdu sentiment analysis with deep learning methods. *IEEE Access* **2021**, *9*, 97803–97812. [CrossRef]

37. Calvo, R.A.; Mac Kim, S. Emotions in text: Dimensional and categorical models. *Comput. Intell.* **2013**, *29*, 527–543. [CrossRef]

38. Buechel, S.; Hahn, U. Emobank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Valencia, Spain, 3 April 2017; pp. 578–585.

39. Mohammad, S.; Bravo-Marquez, F.; Salameh, M.; Kiritchenko, S. Semeval-2018 task 1: Affect in tweets. In Proceedings of the 12th International Workshop on Semantic Evaluation, New Orleans, LA, USA, 5–6 June 2018; pp. 1–17.

40. Zhu, S.; Li, S.; Zhou, G. Adversarial attention modeling for multi-dimensional emotion regression. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 471–480.

41. Wang, J.; Yu, L.C.; Lai, K.R.; Zhang, X. Tree-structured regional CNN-LSTM model for dimensional sentiment analysis. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *28*, 581–591. [CrossRef]

42. Xie, H.; Lin, W.; Lin, S.; Wang, J.; Yu, L.C. A multi-dimensional relation model for dimensional sentiment analysis. *Inf. Sci.* **2021**, *579*, 832–844. [CrossRef]

43. Tang, D.; Wei, F.; Qin, B.; Yang, N.; Liu, T.; Zhou, M. Sentiment embeddings with applications to sentiment analysis. *IEEE Trans. Knowl. Data Eng.* **2015**, *28*, 496–509. [CrossRef]

44. Fu, P.; Lin, Z.; Yuan, F.; Wang, W.; Meng, D. Learning sentiment-specific word embedding via global sentiment representation. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.

45. Yu, L.C.; Wang, J.; Lai, K.R.; Zhang, X. Refining word embeddings using intensity scores for sentiment analysis. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2017**, *26*, 671–681. [CrossRef]

46. Basiri, M.E.; Nemati, S.; Abdar, M.; Cambria, E.; Acharya, U.R. ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. *Future Gener. Comput. Syst.* **2021**, *115*, 279–294. [CrossRef]

47. Kamyab, M.; Liu, G.; Adjeisah, M. Attention-Based CNN and Bi-LSTM Model Based on TF-IDF and GloVe Word Embedding for Sentiment Analysis. *Appl. Sci.* **2021**, *11*, 11255. [CrossRef]

48. Liao, W.; Zhou, J.; Wang, Y.; Yin, Y.; Zhang, X. Fine-grained attention-based phrase-aware network for aspect-level sentiment analysis. *Artif. Intell. Rev.* **2021**, 1–20. [CrossRef]

49. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 3111–3119.

50. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.

51. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]

52. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.

53. Jain, P.K.; Saravanan, V.; Pamula, R. A hybrid CNN-LSTM: A deep learning approach for consumer sentiment analysis using qualitative user-generated contents. *Trans. Asian-Low-Resour. Lang. Inf. Process.* **2021**, *20*, 1–15. [CrossRef]

54. Elzayady, H.; Badran, K.M.; Salama, G.I. Arabic Opinion Mining Using Combined CNN-LSTM Models. *Int. J. Intell. Syst. Appl.* **2020**, *12*, 25–36. [CrossRef]

55. Li, W.; Zhu, L.; Shi, Y.; Guo, K.; Cambria, E. User reviews: Sentiment analysis using lexicon integrated two-channel CNN-LSTM family models. *Appl. Soft Comput.* **2020**, *94*, 106435. [CrossRef]

56. Maas, A.L.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011*; Association for Computational Linguistics: Portland, OR, USA, 2011; pp. 142–150.