applied sciences

MDPI

Article

Attention-Based RU-BiLSTM Sentiment Analysis Model for Roman Urdu

Bilal Ahmed Chandio [1], Ali Shariq Imran [2,*], Maheen Bakhtyar [1], Sher Muhammad Daudpota [3] and Junaid Baber [1,4]

[1] Department of Computer Science and Information Technology, University of Balochistan, Quetta 87300, Pakistan; bilal.csit@um.uob.edu.pk (B.A.C.); maheen.bakhtyar@um.uob.edu.pk (M.B.); junaid.baber@univ-grenoble-alpes.fr (J.B.)
[2] Department of Computer Science (IDI), Norwegian University of Science and Technology (NTNU), 2815 Gjøvik, Norway
[3] Department of Computer Science, Sukkur IBA University, Sukkur 65200, Pakistan; sher@iba-suk.edu.pk
[4] Laboratoire d'Informatique de Grenoble, Université Grenoble Alpes, 38401 Grenoble, France
* Correspondence: ali.imran@ntnu.no

Abstract: Deep neural networks have emerged as a leading approach towards handling many natural language processing (NLP) tasks. Deep networks initially conquered the problems of computer vision. However, dealing with sequential data such as text and sound was a nightmare for such networks as traditional deep networks are not reliable in preserving contextual information. This may not harm the results in the case of image processing where we do not care about the sequence, but when we consider the data collected from text for processing, such networks may trigger disastrous results. Moreover, establishing sentence semantics in a colloquial text such as Roman Urdu is a challenge. Additionally, the sparsity and high dimensionality of data in such informal text have encountered a significant challenge for building sentence semantics. To overcome this problem, we propose a deep recurrent architecture RU-BiLSTM based on bidirectional LSTM (BiLSTM) coupled with word embedding and an attention mechanism for sentiment analysis of Roman Urdu. Our proposed model uses the bidirectional LSTM to preserve the context in both directions and the attention mechanism to concentrate on more important features. Eventually, the last dense softmax output layer is used to acquire the binary and ternary classification results. We empirically evaluated our model on two available datasets of Roman Urdu, i.e., RUECD and RUSA-19. Our proposed model outperformed the baseline models on many grounds, and a significant improvement of 6% to 8% is achieved over baseline models.

Keywords: sentiment analysis; deep learning; Roman Urdu; neural networks; LSTM; attention networks; text processing; natural language processing

check for updates

Citation: Chandio, B.A.; Imran, A.S.; Bakhtiar, M.; Daudpota, S.M.; Baber, J. Attention-Based RU-BiLSTM Sentiment Analysis Model for Roman Urdu. *Appl. Sci.* **2022**, *12*, 3641. https://doi.org/10.3390/app12073641

Academic Editor: Kuei-Hu Chang

Received: 15 February 2022
Accepted: 29 March 2022
Published: 4 April 2022

1. Introduction

Sentiment analysis is a branch of "text classification" and one of the eminent research areas in the field of natural language processing (NLP) [1]. Sentiment analysis can be performed at three levels, i.e., the aspect level, the sentence level, and the document level. Since the inception of social media and other user centric platforms, people began to share their reviews and sentiments against an entity, product, person, or an organisation [2]. Sentiment classification is also deliberated as automatically classifying text documents according to the predefined target polarity [3]. The polarity may be positive, negative, or neutral. Sentiment analysis is also defined as a branch of text classification, which is a field of study that analyses people's opinions, sentiments, appraisals, attitudes, and emotions toward entities and their attributes expressed in written text. Sentiment analysis has been found to be a powerful tool for businesses, governments, and researchers to extract and analyse public mood and views, to gain business insight, and to make better decisions [4,5].

In E-commerce, decision makers are always interested in obtaining feedback from a user so as to trigger the right decision at the right time. In Pakistan and its sub-continent, people mostly express their reviews in Roman Urdu, which is overwhelmingly increasing on the web. Roman Urdu is basically an amalgamation of English alphabets and Urdu. In other words, English alphabets are used to express Urdu, creating Roman Urdu. For example, the word "Justice" in Roman Urdu is equivalent to "Insaf". The role of Roman Urdu in multilingual information retrieval was investigated in a regional study, and it was revealed that Internet users in Pakistan mostly prefer Roman Urdu over conventional nastaliq Urdu script [6]. In the next section, the structure of Roman Urdu is discussed in detail.

The astronomical rise in the amount of online data regarding product services, events, issues, and celebrities has triggered the need to find an automated way to comprehend user reviews. Sentiment analysis, a dedicated task, has emerged as a promising area of NLP in this regard. One of the most compelling reasons to use sentiment analysis widely is that it greatly aids businesses in understanding consumer needs and in formulating critical modifications in marketing and business strategies to improve user experience [7–9].

For data analysts, it is almost impossible to comprehend the amount of data manually. E-commerce systems, if powered by the state-of-the-art sentiment analysis, may yield excellent results. Moreover, sentiment analysis also helps assess other systems such as recommendation systems, hate speech detection, and spam detection, among others. In E-commerce, user reviews play vital roles in decision-making so as to ameliorate business practices. These reviews may include the utility, pricing, promotion, unique features, discounting, delivery, service, organisation, individual, or any other aspect of the experience process with the item. Interpreting the sentiments of the users refers to sentiment analysis and is one of the most active research area in NLP [10].

Contemporary advancements in sentiment analysis approaches based on machine learning and deep learning have greatly improved the performance of business intelligence [11], and scientific and academic applications. During the last few years, the E-commerce sector in Pakistan has grown at a great pace in which the leading E-commerce giant Daraz.pk has played a vital role. The masses in Pakistan and its sub-continent mostly express their reviews/sentiments in Roman Urdu. Owing to the colloquial nature of this text, it is very much challenging to perform sentiment classification. Moreover, there exists no grammar or spelling rules for Roman Urdu, and for a single word, there exists a number of variants, which makes it very much informal. However, in this region, this informal text is the first choice used to express the sentiments of the user.

This paper focuses on Roman Urdu reviews that were scrapped from the famous E-commerce giant Daraz.pk and from Twitter. There were a total 26,824 reviews, which were annotated by three different domain experts into three classes: positive, negative, and neutral. Moreover, another Roman Urdu Dataset RUSA-19 by [12] has also been experimented with for the sentiment classification task.

1.1. Problem Statement

In NLP, a Roman Urdu sentiment analysis has always been a challenging task owing to the colloquial nature of text written in Roman Urdu. While processing a text for analysis, creating a feature vector is an important step. A good feature vector would yield good results, and a poor feature vector will eventually trigger disastrous results. The traditional bag-of-words model or vector space model for feature extraction is not sufficient for dealing with such informal text.

The authors of [13] claimed an accuracy of almost 60% on the Roman Urdu Dataset using the vector space model along with the TF-IDF weighting scheme with an SVM classifier, which should be ameliorated. Among the schemes for sentiment analysis, the n-gram statistical or rule-based methods are the most commonly used. Moreover, there exists no universal scheme of text classification for every kind of text [3]. Most of the text classification models rely on target classes without considering the relationship between word features. Beside some stunning features, these traditional models have certain drawbacks that restrict them when attempting to accurately classify the text.

1.2. Significance of the Study

Artificial neural networks (ANN) have brought about a revolution not only in computer vision and speech processing but also in the field of NLP [14]. The state-of-the-art RNN, LSTM, and BiLSTM for Roman Urdu sentiment analysis is itself a significant study in assessing the power of neural networks. Prior to this, very few studies experimented with Roman Urdu text on neural nets. In contrast, our sequential modelling approach of LSTM and BiLSTM along with an attention mechanism is a novel approach to solving the Roman Urdu Sentiment Analysis puzzle. We empirically evaluated that our scheme outperformed all existing schemes in classifying Roman Urdu sentiments. For a better evaluation, we used two different Roman Urdu datasets with 26,824 and 10,015 number of documented instances in each dataset. The datasets are publicly available for future experimentation (https://github.com/bilalbaloch1/RU-BiSLTMS) (accesed on 14 February 2022).

1.3. Objectives of the Study

This research study has the following objectives:

1. This research study exploits the use of LSTM and BiLSTM for a sentiment analysis of Roman Urdu text. The architecture of this type of RNN is aimed at handling the sequential modelling tasks;
2. It investigates the capability of LSTM and BiLSTM along with the attention mechanism in capturing the contextual information by experimenting with long and short user reviews; and
3. it empirically evaluates three different neural word embeddings (word2vec, Glove, and FastText) for Roman Urdu sentiment analysis.

1.4. Contribution of the Study

Following are the contributions of this study:

- We made a new Roman Urdu dataset that contains more than 26,824 labelled instances obtained from Daraz.pk and Twitter and annotated by field experts publicly available.
- This research work provides three different neural word embeddings—word2vec, Glove, and FastText—that were trained on a enormous amount of Roman Urdu data scrapped from different web pages.
- We propose an attention-based BiLSTM model called RU-BiLSM for the sentiment analysis of Roman Urdu.
- In order to evaluate the integrity of generated neural word embeddings, we experimented with six different arrangements of deep neural networks. The comparison of a variety of deep networks arrangements under three different neural word embeddings will give deep insight to researchers into understanding which approach is superior when solving the sentiment analysis problem for Roman Urdu.

2. Related Work

Deep neural networks have gained much prominence in pattern recognition and machine learning problems during the past few years [14]. Deep-learning-based models have also proved remarkable in the field of computer vision and speech recognition [3,15–17]. Moreover, in the field of NLP deep networks, RNN-based networks have specially achieved greater success [3,18]. Sentence modeling is an important task when performing a sentiment analysis. A document-level sentiment analysis is considered a fundamental step and is preferred over sentence-level sentiment classification, because the document level includes more information stored and extracted at the global level rather than the local level. The ability of neural nets to handle the sequences of different lengths has enabled them to achieve remarkable performances in sentiment analysis. The long short-term memory units are notable among these neural nets [19]. In order to learn the distributed sentence representation via neural network, an external domain knowledge is required. For sentence learning representation, the neural network models such as convolutional neural networks (CNN) or RNN are mostly constructed upon input transformed sequences, syntactic parse trees, or pre-trained word vectors [15].

The advent of social media platforms has brought about a revolution not only in web-based technologies but also in social web mining. During the last decade, researchers have empirically evaluated different machine learning and deep learning strategies to identify the polarity of user sentiments and claimed to have made significant improvements in their social platforms such as twitter [2,20,21]. Some researchers are of the opinion that better pre-processing feature selection techniques can offer an outstanding improvement in the performance of sentiment classifiers, whereas others focus more on fine-tuning and on the selection of better classifiers to mitigate this problem [20]. Additionally, the phenomenal success of deep learning approaches for classification tasks has captivated researchers toward opting for neural network prediction-based embedding, activation, and loss criteria along with multi-layer and hybrid deep architectures to achieve the highest classification accuracy [21]. Contrary to this, many researchers still prefer machine-learning-based approaches for text classification tasks along with traditional feature engineering techniques [13]. On the other hand, some researchers have attained superior performance by utilising transfer learning strategies on already trained language models in resource-rich languages [22]. Roman Urdu being resource starved has no such pre-trained models or word embedding to be utilised for transfer learning or any other futuristic approach.

Nonetheless, an ample amount of research has been conducted on many famous languages such as English, German, Chinese, and Arabic [23–27]. In contrast, very limited research work has been conducted on South-Asian languages, more specifically on Roman Urdu [7,12,13,28]. The lack of language resources, such a benchmark dataset and pre-trained neural embedding, has been a primary bottleneck in mitigating the sentiment analysis problem [22].

Traditionally, the bag-of-words model or vector space model is used for sentence modeling and feature extraction and suffers from the curse of dimensionality [15]. It was further argued that the CNNs and RNNs have emerged as two widely used architectures. The CNN, having power to capture the local correlations of spatial and temporal structures, has achieved greater success not only in computer vision and speech recognition but also in NLP tasks [15,16]. CNN has shown good performance in extracting textual features. While dealing with text data, the CNN found excellence in extracting n-gram features using convolutional filters. However, in the Roman Urdu case, CNN did not find significance in extracting textual features. On the other hand, the RNN has shown remarkable performance for sequential modeling tasks [29] contrary to CNN.

The RNN's capability to deal with arbitrary length of sequences [19] enables it to be the first choice in sequential modeling tasks. The RNN such as LSTM has shown superior performance while preserving contextual information. Hence, the LSTM was found to be much more reliable in capturing spatial and temporal dependencies [29]. In contrast, the traditional feed-forward network fails to preserve the contextual information and cannot handle long-term dependencies. Owing to the high dimensionality of the sequential data (such as text), it may trigger a big challenge, and the solution to such spatial data is BiLSTM, which overcomes many long and short dependencies. Many researchers have utilised the fusion of both CNN and RNN for text classification. The authors of [3] have proposed an attention-based BiLSTM with a convolutional layer scheme called AC-BiLSTM for sentiment classification. Their proposed recipe uses convolutional layers to extract higher level features, which are further fed into LSTM hidden units to learn long-term dependencies. They concluded that the convolutional window size [30] and stride size affect the performance, whereas BiLSTM and attention mechanism have more significant effects on the classification accuracy than convolutional layers.

Our proposed scheme is also based on BiLSTM, except we omitted the use of CNN layers owing to the lack of a significant impact on classification accuracy. The authors of [15] also amalgamated CNN and RNN and proposed a scheme named C-LSTM without an attention mechanism for sentence representation and text classification. In a recent study on text classification, it was argued that most of the representation models learn insufficient structural information or depend on already trained structures, which results in performance degradation [31]. A sandwich neural network (SNN) was proposed and designed to capture local semantics and global structural representations. Their proposed scheme was based on a knowledge attention mechanism to capture the information on both the instance level and the corpus level. Their attention mechanism was based on encoder RNN and decoder RNN which show the significance of such a neural network.

In text classification, dealing with a sequences of long-time-stamp is a major challenge [32]. Rao et al. [19] have also identified this problem in traditional neural networks and proposed the LSTM-based model SR-LSTM. Their two-layer LSTM model learns the semantics of sentences in the first layer and, in second layer, establishes the relationship among document sentences. In a research study conducting a sentiment analysis of Amazon product reviews, different machine learning algorithms (Naive Bayes, Support Vector Machine, and Maximum Entropy) were experimented. The learning classifiers were trained on unigrams and weighted unigrams. It was identified that machine learning algorithms work well on weighted unigrams and that SVM has shown better performances [33]. Recently, a personalised sentiment classification approach that considers additional attributes such as product information and user-text was introduced [34]. To extract implicit information, three interactive attribute encoders were used in addition to the local text encoder: user–product, user–text, and product–text encoders. They empirically evaluated three different datasets: IMDB, Yelp and Amazon Reviews. Then, they claimed to have achieved a significant improvement in performance for the proposed approach. In another research study, the authors focused on a sentence-level sentiment analysis. They proposed a sentence-to-sentence attention network (S2SAN) [35]. They were of the view that a word representation approach for sentiment analysis overlooks the importance of each sentence, which can be dealt with using a sentence-level attention mechanism. It was identified that deep learning classifiers such as CNN, RNN, and LSTM embedded within a sentence attention framework yields good results.

Elfaik et al. [36] has proposed Bidirectional LSTM for Arabic sentiment analyses. Arabic being a morphologically rich and complex language has presented challenges for researchers to undertake sentiment analyses. The experiments were conducted on six benchmark Arabic datasets, which revealed that the proposed model outperformed the state-of-the art machine learning and deep learning models. In a study conducting a sentiment analysis of Chinese textual data, it was identified that sentence semantics play significant roles [37]. To assess the sentiment tendency of Chinese texts, this research offered a scalable multi-channel dilated combined architecture of the convolutional neural network and bidirectional long short-term memory (CNN–BiLSTM) model with an attention mechanism. The study was aimed at exploiting both the original context features and high level features using multi-scale dilated architecture.

In contrast to deep learning, some researchers have employed lexical knowledge graph-based techniques. The majority of recent Lexical sentiment analysis (LSA) methods have relied on supervised learning techniques applied to corpus-based statistics, which need a lot of training data, a lot of time, and a lot of statistical corpora. Other research have used unsupervised and lexicon-based algorithms to match target words in a lexical knowledge base (KB) with seed words in a sentiment lexicon but have run into issues with affective concept coverage and connectedness. Fares et al. [38] has introduced an LSA-based unsupervised approach LISA, a knowledge graph-based LSA framework that is unsupervised at the word level. It computes and propagates affective scores in a lexical-affective graph (LAG) formed by linking a conventional lexical KB such as WordNet with a trustworthy affect KB such as WordNet-Affect Hierarchy using different variations of shortest path graph navigation algorithms.

Sequence-to-sequence deep learning models are a class of Recurrent Neural Networks (RNNs) that are used for different language problems such as machine translation, text summarisation, question answering, and text generation. Since their inception, deep generative models saw a boom in popularity among deep learning researchers. These models are used to generate new synthetic data such as photos, movies, and texts by fitting approximate distributions to the data [39]. The generative models are also utilised for drug discovery and molecular design. Several ways of developing algorithmic music composers have been explored in recent years. The majority of current methods focus on making music that looks to be technically right or listener-friendly and captivating. However, few approaches have been developed to create music that communicates human emotions based on sentiments. Abboud et al. [40] has introduced an algorithmic approach for autonomous music sentiment-based expression and composition, namely MUSEC, that may understand an extensible set of six primary human emotions (e.g., anger, fear, joy, love, sadness, and surprise) expressed by a MIDI musical file and then composes (creates) new, polyphonic (pseudo) thematic and diversified musical pieces that express these emotions. Some researchers have also employed generative models for sentiment detection of stock messages of investors. The generative models converts the messages to emotion vectors during the training process. Subsequently, these emotion vectors are fed to the classifier. Their experimental results reveal that generative models are effective for short-messages sentiment classification. Duan et al. [41] has utilised a semi-supervised approach for the sentiment analysis of stock messages of investors. Aspect-level sentiment analysis is a fine-grained sentiment analysis technique that aims to determine the sentiment polarity of a given opinion target conveyed. An aspect-level sentiment analysis approach was introduced, titled as M-IAN, to capture the interactions between target and context using a deep interactive memory network [42].

Word representation plays a vital role in the text classification task. In a series of experiments on benchmark datasets using CNN with word2vec, Kim achieved a remarkable accuracy [16]. It was identified that, with a little fine-tuning of the CNN hyperparameters, a remarkable performance can be achieved. Kim was of the view that pre-trained word vectors could better help deep networks improve their performance. A recent study on aspect-based sentiment analysis has proposed a deep learning model NA-DLSTM. They deliberated that sentiment analysis and aspect-based sentiment analysis can be differentiated in a way that the former considers the whole document whereas the later analyses and categorises the text into various aspects and assigns the polarity to each aspect. The study was aimed at developing a context-aware aspect-based SA model, experimenting on the Rest14, Rest15, and Rest16 dataset [43].

An important task while performing the sentiment analysis is feature selection, which suffers in the presence of uneven class distribution of the datasets [44,45]. In text classification, usually two classes are involved: the positive class (the class of interest) and the negative class, which is mostly over-represented. This over-representation is known as the class imbalance problem [46]. This imbalanced class distribution allows for the redundant, noisy and erroneous features to gain more weight. An earlier study of imbalanced data by [42,47,48] also identified that imbalanced class distribution poses a significant challenge to the classifier's performance. To overcome this problem, they proposed the Adaboost algorithm. However, this uneven distribution was still a challenge while selecting the features. Both of our Roman Urdu datasets also face the same issue of imbalanced class distribution, which triggers a performance downgrade. There exists no pre-trained word vectors for Roman Urdu. Assessing the need to enrich this resource-starved language, we introduced for the very first time three different word embedding schemes: word2vec, Glove, and FastText.

2.1. An Insight into Roman Urdu Sentiment Analysis

In a systematic literature review (SLR) on Roman Urdu sentiment analysis, it was identified that the Urdu language lacks standardised datasets for sentiment analysis, which is critical [49]. Furthermore, pre-processing and feature selection mechanisms should be designed in accordance with Urdu language's peculiarities. In a recent study on lexical normalisation on Roman Urdu, a phonetic algorithm-based approach TERUN (Transliteration based Encoding for Roman Urdu text Normalisation) was proposed [50]. TERUN uses features of Roman Hindi/Urdu linguistics to convert lexically variable terms to their canonical forms. Transliteration-based encoder, filter module, and hash code ranker are the three interconnected modules. For a single Roman Hindi/Urdu word, the encoder creates all possible hash-codes. The third module ranks the filtered hash-codes depending on their relevancy, while the fourth component filters the irrelevant codes. The goal of this research is not only to normalise the text but also to see how it affects text classification.

Recently, a novel term-weighting technique was introduced known as discriminative feature spamming method (DFST), which is a unique word weighting technique for Roman Urdu sentiment analysis that selects distinctive terms based on a term utility criteria (TUC) and then spams them to improve their discriminative strength [28]. The experiments were conducted on 11,000 Roman Urdu reviews and a custom tokeniser was built to improve the accuracy. In an earlier study Ayesha et al. [51] collected Roman Urdu product reviews for sentiment analysis. They experimented with three machine learning schemes: Naive Bayes, SVM, and Logistic Regression. They revealed that SVM outperforms the rest of the classifiers. Likewise, Noor et al. [13] also proposed SVM for this task. They collected 20,000 product reviews from Daraz.pk and experimented with different SVM kernels. An accuracy of 60% was reported under linear SVM kernel and bag-of-word features.

In 2016, Bilal et al. [52] experimented with a very small set of Roman Urdu reviews containing 150 positive and 150 negative reviews. They employed three machine learning classifiers including Naive Bayes, Decision Tree, and KNN using WEKA tool. It was identified that Naive Bayes showed a better performance than Decision Tree and KNN in terms of accuracy, precision, recall, and F-measure. In another attempt to perform a Roman Urdu sentiment analysis, reviews were collected from different automobile websites for a binary classification task. Arif et al. [53] empirically evaluated ten different classifiers: SVM, KNN, Decision Tree, Passive Aggressive, Ensemble classifier, Perceptron, SSGD, Naive Bayes, Ridge classifier, and nearest centroid. In their experimental settings, they employed three different feature representation schemes i.e., TF, TF-IDF, and Hashing vectoriser, and three feature selection schemes, i.e., Chi-Squared, IG, and MI. Their empirical evaluation of diverse machine learning classifiers determined that SVM outperforms the rest of the classifiers. Naqvi et al. [54] conducted a study using Naive Bayes, Logistic Regression, LSTM, and CNN on Roman Urdu News categorisation. The goal of the research was to categorise the news into five groups (health, business, technology, sports, and international). For lexical normalisation and noise reduction in the corpus, they used a phonetic algorithm. Naive Bayes beats the other classifiers, according to the findings. Chandio et al. [55] has recently experimented with a bunch of machine learning algorithms for the sentiment analysis of Roman Urdu and with their proposed SVM.

Some researchers have also utilised deep neural networks for the sentiment classification of Roman Urdu. Mehmood et al. [12] created a new Roman Urdu corpus RUSA-19 containing 10,012 reviews. The reviews belongs to six different domains: (i) drama, movies, and talk shows; (ii) food and recipes; (iii) politics; (iv) apps, blogs, forums, and gadgets; and (v) sports gathered from numerous websites on social media. They proposed a deep network RCNN for deep sentiment detection and experimented with newly created dataset. Their proposed deep model was based on recurrent layers and CNN layers. However, it is observed by several researchers that CNN is better at handling non-sequential data, whereas for sequential data, CNN has not shown satisfactory results. Ghulam et al. [56] conducted a deep-learning-based comparison on Roman Urdu for the sentiment analysis problem. Different machine learning classifiers were empirically examined in conjunction with the suggested deep learning model LSTM. It was discovered that the deep learning model outperforms the machine learning models. In a recent study, a multi-channel hybrid approach that proposes a method based on BiGRU and word embeddings was utilised [7]. However, their dataset was too small, containing just 3241 sentiment reviews, and is not yet publicly available. Rizwan et al. [57] has conducted a research study on hate speech detection on Roman Urdu and employed static CNN with different word embedding schemes.

Most of the previous work reveals that Roman Urdu lacks a benchmark dataset and feature representation scheme. We tried to address this problem by introducing the largest ever Roman Urdu dataset RUECD and three different neural word embedding schemes.

2.2. Long Short-Term Memory (LSTM) and Backpropagation

The pioneers of LSTM [58] determined that storing information of previous outputs via recurrent back-propagation for an extended period of time takes much time. It was identified that, in contrast to traditional recurrent learning, back-propagation through time (BPTT), and other learning strategies, LSTM can learn more efficiently by truncating the gradient without compromising the efficiency. Solving this gradient descent problem allows LSTM to not explode and destroy the gradient.

In a comparative study on LSTM variants for different benchmark problems—acoustic modeling, handwriting recognition, and polyphonic music modeling—the classic vanilla LSTM proved to be better against its variants [29]. However, two hyperparameters, i.e., coupling the input and removing the peehole connections, have helped LSTM in cutting the processing cost without degrading the performance.

2.3. Theoretical and Conceptual Framework

The literature review clearly depicts that RNNs, especially the LSTM and BiLSTM, have emerged as a leading approach in text classification problems. We also proposed an LSTM-and-BiLSTM-based approach with an attention mechanism to classify the polarity of Roman Urdu text. The proposed scheme aims to not only overcome the long-term dependencies problem but also establish the semantic relationship between features. Our framework, as depicted in Figure 1, initially performed certain pre-processing tasks on a text corpus and converted them into vector sequences. Subsequently, these text sequences with integer values assigned were fed into our deep neural model to classify the sentiment. The forthcoming section will deliberate the proposed deep learning model and discuss the results.

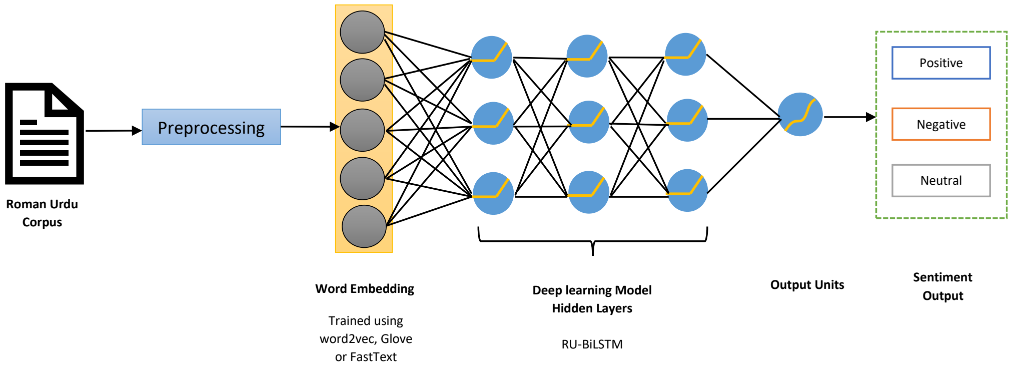


Figure 1. Proposed RU-BiLSTM deep learning framework.

3. Methodology

Roman Urdu is perceived as a symbolic representation of a linguistically-rich but morphologically complex language: Urdu [7]. The nastaliq format is basically a cursive style of writing Urdu that is based on 36 alphabets. In contrast, Roman Urdu relies on English/Latin alphabets and phonetically sounds like Urdu. Owing to less morphological support from English/Latin alphabets, Roman Urdu confronts many challenges. People in South-Asia mostly prefer to communicate on social platforms via Roman Urdu, which can easily be typed using a QWERTY keyboard. The increased use of Roman Urdu on the

Internet by a wider Urdu-speaking community has made processing it difficult. The smart processing of Roman Urdu text could be a step forward in developing intelligent systems such as cyber bullying and hate speech detection, recommendation systems, neural dialog generation, and text summarisation and generation.

The major bottleneck in processing Roman Urdu is the colloquial nature of text written in Roman Urdu. This inconsistent representation of text has posed a serious challenge for researchers to mitigate this problem. In other words, no rules exist for Roman Urdu because of spelling variations. Everyone has their own spelling criteria for expressing Roman Urdu; hence, one word has infinite spelling variations. Roman Urdu is just a representation scheme for Urdu; it does not inherit any rules. Many researchers have, at first, proposed a rule-based scheme to standardise Roman Urdu text, but unfortunately, it has suffered owing to the inconsistent nature of the text. Such inconsistency leads to low performance when selecting the features and semantically weakens the model. We tried to overcome this issue by employing three different neural word embedding so as to semantically empower the proposed model because formulating a pre-processing scheme for a non-standardised text does not yield fruitful results. A word, for instance, "beautiful", can be represented in Roman Urdu as multiple variants such as "khoobsorat", "khuobsrt", "khubsuret", and "khobsorat". These variations are basically caused by different pronunciations or accents. It can be observed that, for a single word, there are multiple variants, which are difficult to standardise. A sample of Roman Urdu sentences from three different classes is illustrated in Table 1.

Table 1. Roman Urdu sample sentences with English translations belonging to the three classes.

Sentiment Class	Roman Urdu	English
Positive (1)	Daraz per jo mobile main ny khareeda wo behtareen tha.	The mobile I purchased on Daraz was the best.
Negative (0)	Speaker jo main ny online mangwae thy ki sound quality bohat bury thi or mehnga bhi tha.	The sound quality of the speaker I had ordered online was the worst, and the speaker was expensive too.
Neutral (2)	electronic aalat per jo sale lagi hai, pora mahena chalegi.	The sale that started on electronic equipment is suitable.

Owing to the dynamic nature of Roman Urdu, it cannot be dealt with as an English language even if it uses English alphabets for symbolic representation. Moreover, the semantic and linguistic properties of English and Roman Urdu are different and cannot be mixed. Hence, heterogeneity of words is the major challenge in processing Roman Urdu.

We propose a deep learning recurrent neural network model, RU-BiLSTM, for Roman Urdu sentiment analysis. The model is a combination of input layers of text sequences: an embedding layer; a regularisation dropout layer; two hidden (BiLSTM) units layers with certain regularisations, attention mechanisms, and ReLU activation function; and eventually, a last dense output layer with three outputs, an Adam optimiser, and softmax activation function. In our pipeline, the data are first pre-processed and cleaned. During the pre-processing phase, all kinds of punctuation and English stop words that are mixed in with the Roman Urdu are removed. The cleaned documents are further tokenised and converted into numerical sequences. In other words, each unique word in the corpus is assigned an integer value. Furthermore, these sequences along with encoded labels are fed into an embedding layer of our model. The framework of our pipeline is illustrated in Figure 1. Moreover, a more detailed view of our model is also exhibited in Figure 2, whereby it is evident that BiLSTM encoding is initially applied at the word level and subsequently at the sentence level. Additionally, an attention mechanism is also applied at both the word and sentence levels. The next section will deliberate on each layer of our model RU-BiLSTM.

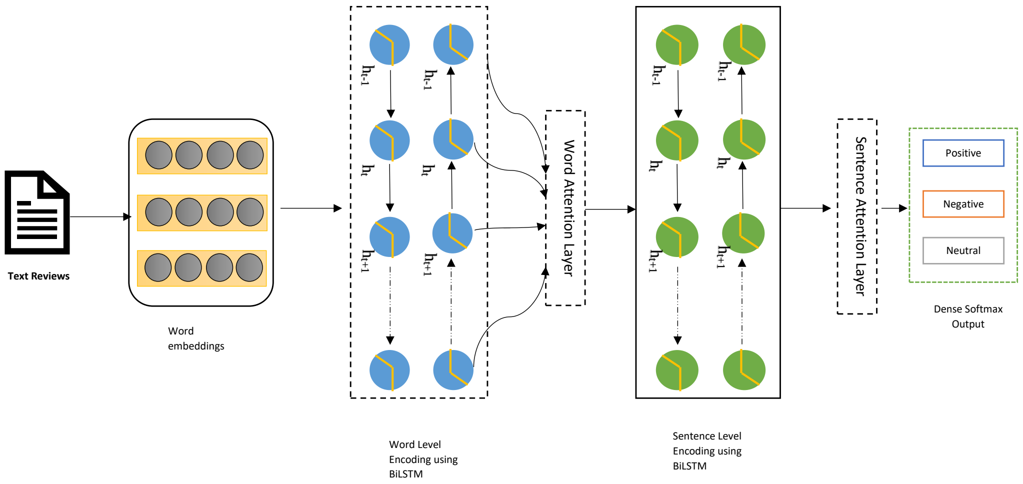


Figure 2. Attention-based bidirectional long short-term memory RU-BiLSTM model.

3.1. Embedding Layer

The primary objective of word embedding is to set lower dimensional dense vectors for the onward layers of neural network. The traditional one-hot encoding tends to be very costly in terms of computation because of too many 0s. The solution to this problem is word embedding, which efficiently represents the text data by reducing the dimensions and by assigning an average learned weight. We employed three famous neural word embedding schemes: word2vec [18], Glove [59], and FastText [60]. The neural word embeddings enable our proposed model RU-BiLSTM to perform in-depth feature extraction to further ameliorate the classification. An enormous amount of Roman Urdu text tuned to one million words crawled from social networking and Roman Urdu news sites is used to create word embeddings for better text representation. The sources of Roman Urdu text data are recorded in Table 2. Before feeding the data to our main model, the input-padded sequences are embedded into the word embedding hidden layer. We provided a document $\mathcal{D}$ having $\mathcal{M}$ number of words, $D = \{x_1, x_2, \ldots x_M\}$, where x_i represents the word in D. We used the embedding matrix $\mathbf{W^k}$ as a lookup table and dictionary for each word in D, where $\mathbf{W^k} \in \mathbb{R}^{d^w|V|}$; here, V is our vocabulary size, which is fixed to 10,000, and $\mathbf{d^w}$ is our embedding size. The weight matrix $\mathbf{W^k}$ is to be learned, whereas $\mathbf{d^w}$ is the hyperparameter to be provided by user. This weight matrix is initially randomly initialised. The aim of word embedding is to shape the x_i *into* p_i, which is basically the product of weight matrix and vector.

$$p_i = W^k \cdot v^i, \tag{1}$$

where v^i is a one-hot encoding vector of vocabulary size $|V|$.

In this one-hot encoding vector at index p_i, the value is 1 and the rest are 0. Eventually, we learn the vector representation for each word that is p_i of desired output dimensions d^w. Hence, our word embedding based on floating-point values would be represented as follows:

$$\mathrm{em} = \{\mathrm{p}_1, \mathrm{p}_2, \ldots, \mathrm{p}_\mathrm{M}\} \tag{2}$$

Table 2. Roman Urdu text sources.

S.No.	Source
1.	www.romanurdu.ythisnews.com (accesed on 14 February 2022)
2.	www.twitter.com (accessed on 14 February 2022)
3.	www.facebook.com (accessed on 14 February 2022)
4.	www.daraz.pk (accesed on 14 February 2022)
5.	www.hamariweb.com (accesed on 14 February 2022)

3.2. Regularisation

Adding some regularisation, such as spatial dropout after the embedding layer, helps restrain the overfitting to some extent. Outputs obtained from the embedding layer are further minimised using a dropout layer. We added a spatial dropout layer to evaluate its affect on accuracy and loss and experienced a slight improvement. We set the dropout to 0.2 as setting up a higher number such as 0.5 or 0.75 leads to a loss of parameters, which causes a decline in performance.

4. Recurrent Neural Network

We empirically evaluated both classic LSTM and its variant BiLSTM. RNN has the power to capture historical information. In other words, RNNs can preserve the previous outputs, whereas the simple feed-forward networks just forward the new input and lack memory. Simple RNNs can only preserve the information of short-time stamps and fail to remember long time stamps. This problem is known as long-term dependencies, which can be dealt with using LSTM. The state-of-the-art LSTM units has the capability to deal with this long-term dependency problem. The recurrent architecture of LSTM can not only preserve the contextual information but also overcome the exploding and vanishing issue of the gradient descent. The gating units of LSTM enables it to control the flow and to decide what to ignore and what to update. The simple RNN as depicted in Figure 3 has a single layer recurrent module with a *tanh* squashing function. Provided that we have an input neuron x_t and hidden output state h_t and the previous hidden output state h_{t-1}, then the simple RNN can mathematically be represented as follows:

$$h^t = g_h(W_i x^t + W_R h^{(t-1)} + b_h) \tag{3}$$

$$y^t = g_y(W_y h^t + b_y) \tag{4}$$

Here, h^t is the hidden output, g_h is a squashing function, W is the weighted matrix, b is a bias, and y^t represents the final output.

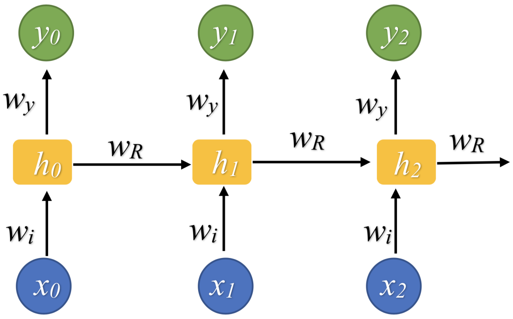


Figure 3. Simple RNN architecture.

The key idea behind *LSTM* is cell states that act like conveyor belts. This conveyor belt transfers the information down in a chain of sequences and passes through different cell states with some minor linear interaction. During this journey, some information is added or removed to the cell states using gates. The classic LSTM consists of four memory blocks, each having a cell state c and three multiplicative units called gates. The gates are input gate i_t, forget gate f_t, and output gate o_t, as illustrated in Figure 4 (source: https://www.edureka.co) (accesed on 14 February 2022). A single memory block of LSTM can mathematically be demonstrated as follows:

$$f^t = \sigma(W_{xf} x_t + W_{hf} h^{t-1} + b_f) \tag{5}$$

$$i^t = \sigma(W_{xi} x_t + W_{hi} h^{t-1} + b_i) \tag{6}$$

$$c_t^{\sim} = \tanh(W_{xc} x_t + W_{hc} h^{t-1} + b_c) \tag{7}$$

Updating the cell state

$$c_t = f_t \otimes c_{t-1} + i_t \otimes c_t^{\sim} \tag{8}$$

$$o_t = \sigma(W_{xo}x_t + W_{hc}h^{t-1} + b_o) \tag{9}$$

$$h_t = o_t \otimes \tanh(c_t) \tag{10}$$

Here, σ represents the sigmoid function, and f_t, i_t, c_t, o_t and h_t show the forget gate, input gate, cell state, output gate, and hidden state, respectively. In contrast, W denotes that the weight matrix corresponds to each separate layer and that b is the bias. The very first step in LSTM is to determine which information is not needed and may be ousted from the cell state. The sigmoid layer, also known as the forget layer, is responsible for this ruling. Second, to determine what new information is to be added, we have two-pass layers: a sigmoid layer called input gate layer, which decides the values to be updated, and a tanh layer, which generates a vector of new candidate values to be stored in cell state $c_t^{\sim}$. Subsequently, the old cell state c_{t-1} is updated with the new cell state c_t, as mentioned in Equation (8). Eventually, we run a sigmoid layer [0, 1] to determine what portion of the cell state should be included as output o_t, which is further element-wise multiplied (denoted with $\otimes$) with a tanh layer having cell state as an argument. The tanh function restricts the values in between -1 and 1. Consequently, we obtain a hidden output h_t.

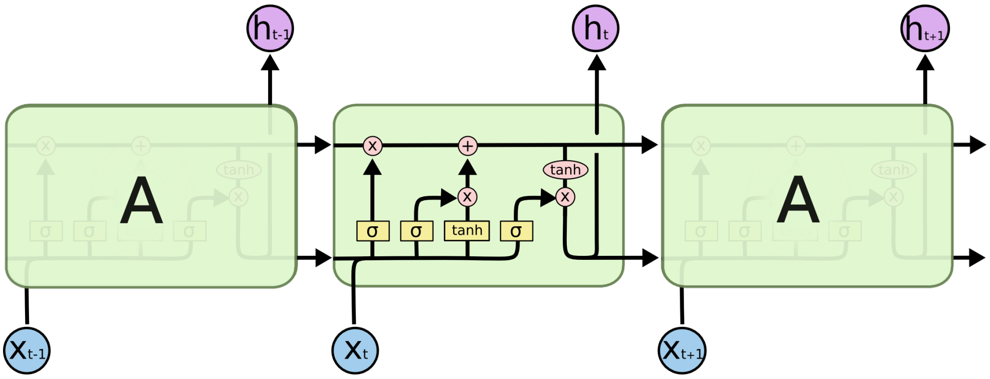


Figure 4. LSTM recurrent architecture (source: https://www.edureka.co) (accesed on 14 February 2022).

4.1. Attention-Based Bidirectional Network

The classic LSTM has one drawback: it only considers the previous outputs. Our BiSLTM powered by the attention mechanism enables us to consider the context in both directions and uses previous and future contexts. For this purpose, two separate hidden bidirectional layers are utilised, which are eventually combined to a single output layer and fed into attention layer. Our bidirectional network consists of two hidden LSTM layers that are forward and backward layers denoted as L_f and L_b, respectively. The two layers traverse the sequences of features S_C from 1 to n and vice versa. The network is represented mathematically as in Equations (11) and (12). The BiLSTM layer determines the annotations of words in both directions and eventually summarises the information. The attention mechanism allows to pay attention towards the word, which contributes more to the specific class of sentiment text. We used attention mechanism based on [61]. The attention layer gives more importance to key features and discards non-important features. The acquired features are further processed by the dense softmax layer, which eventually generates the final output.

$$\overrightarrow{h_f} = \overrightarrow{L_f}(S_C)_n \ , n \in [1, 100] \tag{11}$$

$$\overleftarrow{h_b} = \overleftarrow{L_b}(S_C)_n \ , n \in [100, 1] \tag{12}$$

Attention Mechanism

The attention mechanism is applied to each hidden output $\mathbf{h_t}$ obtained from the LSTM layers, so as to assign different weights to words that contribute differently. A weighted combination of all hidden states is a popular approach of assigning various weights to distinct words in a phrase. The attention model is aimed at computing the context vector as follows:

$$u_t = \tanh\left(W \cdot h_t + b\right) \tag{13}$$

$$\alpha_t = \frac{\exp(u_t^T u_w)}{\sum_t \exp\left(u_t^T u_w\right)} \tag{14}$$

$$c = \sum_t^T \alpha_t h_t \tag{15}$$

where u_t is a hidden representation of h_t and u_w depicts the context vector that is randomly initialised and subsequently learned during the training phase. In order to ascertain the importance of word u_t, the similarity is computed between u_t and u_w. T is the total number of time steps in the input sequence, and α_t is a weight computed at each time step t for each state h_t . The weightings α_t are then computed as shown in Equation (14). By applying a weighted sum to these importance weights, they are eventually aggregated into c as formulated in Equation (15). c represents the vector and summarises all of the information of words in the text review.

4.2. Learning the Model

In order to train our model for the text classification task, the BiLSTM hidden layer is placed at the final time-stamp to obtain the hidden state output for the text representation and, on top of the stack, a dense output layer with softmax activation function. We trained our model to also curtail the loss at the end of each iteration; for this purpose, we used two cross-entropy loss functions separately, i.e., binary cross-entropy and categorical cross-entropy. Moreover, we used an NAdam optimiser (variant of adam) to enable the model to learn quickly and provided a set of training samples x^j and its target labels $y^j \in \{1, 2, \ldots, d\}$, where d is the number of possible document labels. Moreover, the approximated tags likelihood is $y_l^{\epsilon(j)} \in [0, 1]$ for the labels $l = \{1, 2, \ldots, d\}$. The loss function can mathematically be represented as follows:

$$L(x^j, y^j) = -\sum_{c=0}^{N} z_k \ln(y_k^{\epsilon}, c) \tag{16}$$

Here, N is the number of classes, i.e., positive (1), negative (2), and neutral (0); ln is the natural log; z is a binary measure to indicate that whether the class label is a true classification or false for the kth observation; and y^{ϵ} is the predicted likelihood observation k of class c.

4.3. Learning the Optimal Model Weights

In order to learn the optimal weights for the proposed model, we empirically evaluated a set of learning optimisers on a suitable number of epochs. The deep learning models mostly over-fit the training set; to avoid overfitting, an early stopping mechanism is used to restrict the model on fewer number of epochs. Additionally, a dropout layer, which is tuned to 0.5, was also utilised to learn the optimal weights. It can be clearly observed in Figure 5 that up to the fifth epoch, the accuracy improved; afterwards, there is mostly a decline. The experimental evaluation reveals that a higher number of losses is incurred if we keep training the model; therefore, we finalised the weights, which the model learned after five epochs.

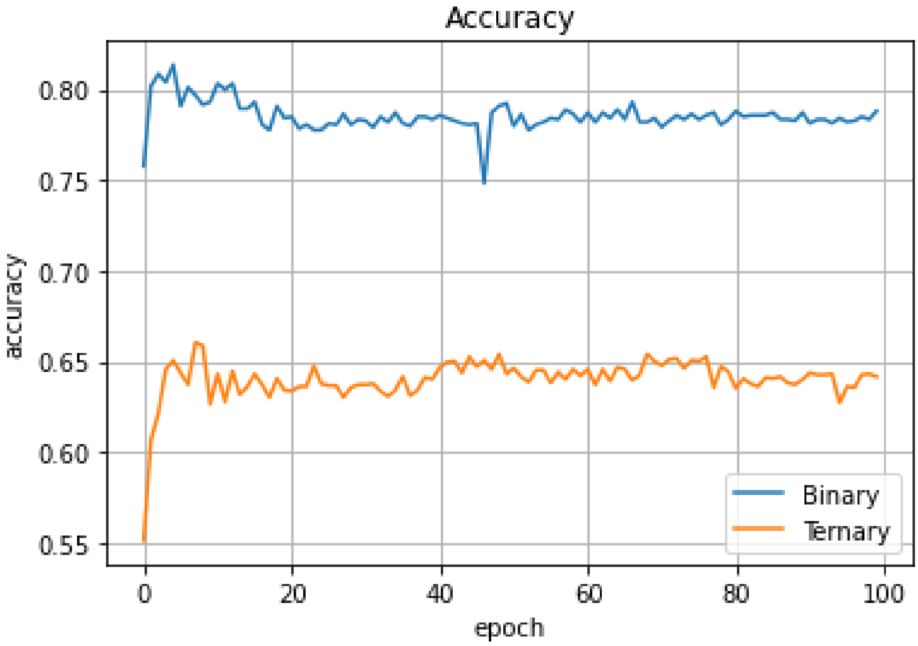


Figure 5. Validation accuracy comparison of binary and ternary classification on the RUECD Dataset.

5. Experiments

In our research framework for Roman Urdu sentiment analysis, we empirically evaluated two Roman Urdu datasets/corpora: Roman Urdu E-commerce dataset (RUECD) (26,824 reviews) and RUSA-19 (10,021 reviews) [12]. In this section, the experimental setup, corpus generation, and certain baseline methods with discussions are described.

5.1. Experimental Setup

We evaluated our newly created Roman Urdu E-commerce Dataset (RUECD) annotated for three classes: positive, negative, and neutral. Moreover, the RUSA-19 [12] dataset is also experimented on using our proposed model. The sentiment analysis task is performed on the model for both binary and ternary classification so as to investigate the potential of the proposed model. The binary classification aimed at classifying the positive and negative reviews, whereas the ternary classification will target three classes, i.e., positive, negative, and neutral. We performed a series of deep learning experiments for both classification tasks and will be discussed in subsequent sections.

In our experimental setting, we used the tensorflow keras deep learning library in python. Moreover, sckit-learn was used to split the training and testing data by random sampling. The promising performance of three neural word embeddings, i.e., word2vec, Glove, and FastText, in different NLP tasks compelled us to evaluate its integrity in our empirical configuration.

5.2. Corpus Generation and Statistics

In order to create a benchmark corpus for Roman Urdu, we scraped product reviews from Daraz.pk, Twitter, and other famous E-commerce websites. In total, a 26,824 set of user reviews was collected and manually annotated by three experts who have good knowledge of Urdu and, more specifically, Roman Urdu. The annotators were asked to label data as per the annotation guidelines furnished to them.

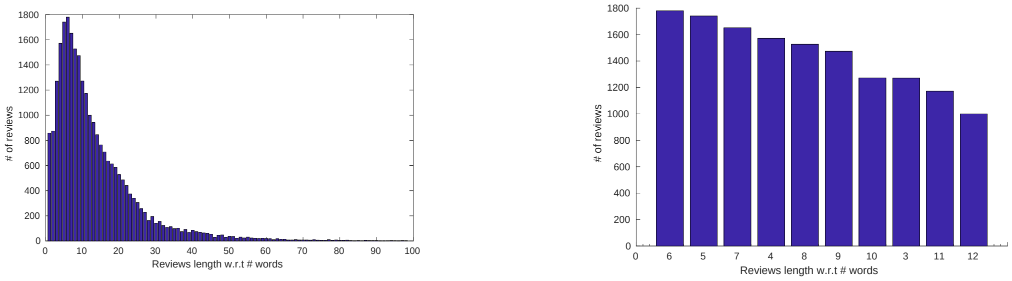
5.2.1. Dataset Properties

The Roman Urdu E-commerce dataset (RUECD) was annotated for the sentiment analysis task and was categorised into three classes, i.e., positive, negative, and neutral. The RUECD dataset includes 26,824 user reviews with 9833 positive, 8252 negative, and

8739 neutral reviews, as illustrated in Table 3. In the dataset, the shortest review has a length of 1 word and the longest review has a length of 296 words. The average length of reviews was computed as 74 words, with standard deviation of 54. The median estimated was 66, which is almost at par with the mean. This distribution of reviews, as illustrated in Figure 6, depicts that most reviews are between 1 to 12 words, with long reviews being very few and easy to classify but with short reviews being a little tricky when deciding the sentiment class.

Table 3. RUECD dataset description; there are a total of 26, 824 reviews divided into two sets, i.e., training set and test set.

Sentiment Class	**Training Set**	**Test Set**	$\sum$
Negative (0)	6601	1651	8252
Positive (1)	7866	1967	9833
Neutral (2)	6991	1748	8739
$\sum$	21,458	5366	26,824



(**a**) (**b**)

Figure 6. The distribution of reviews length: (**a**) shows the overall distribution of length from 1 to 100, and (**b**) shows the top 10 review lengths.

5.2.2. Annotation Process

The annotation process was initiated by designating three annotators—A, B, and C—to manually annotate user reviews separately. Each annotator was assigned a user review to label to one of the three classes (positive, negative, or neutral) according to the furnished annotation guidelines. In order to address the efficacy of annotation process, a sample of 300 reviews (annotated by A and B) was evaluated for conflicting pairs. In case of any disagreement, the third annotator C was assigned to label the conflicting review. Moreover, an inter-annotator agreement (IIA) was established for the whole corpus. For the whole corpus, we achieved an Inter-Annotator Agreement (IAA) of 72.45% and a Cohen Kappa value of 0.58.

The Roman Urdu RUECD dataset, which consists of 26,824 text reviews, has the following class distribution: 9833 positive, 8252 negative, and 8739 neutral reviews. The dataset was split into training and test data in a 80%:20% ratio. Subsequently, the training data were further split for validation data in a ratio 80%:20%.

The RUSA-19 [12] dataset contained 10,015 user reviews with 3000 positive, 4000 negative, and 3000 neutral instances. This corpus was also split in the same manner as the above.

5.3. Evaluation Metrics and Parameter Fine-Tuning

In order to ascertain the classification performance of our model, we used the precision, recall, F1-measure, and accuracy as evaluation metrics. Moreover, the loss was also recorded to estimate the error in classification, as discussed earlier. The training process starts with feature extraction by embedding the input sequences x^t into vectors p. The dimensions for word embedding was set to 300, and 128 BiLSTM memory units were used along with the default tanh activation function. The recurrent dropout for BiLSTM was adjusted to 0.2, and return sequences were set to true. On top of the stack, we used a dense layer with softmax activation having two and three outputs for binary and ternary classification, respectively.

A batch size of 32 was used for learning purposes. Furthermore, a back-propagation algorithm was used along with different optimisation functions such as RMSprop and adam. The learning rate was tuned to 0.001. By tuning the hyperparameters, such as embedding dimensions and optimisation functions, and learning rate, we tried to curtail the loss and to ameliorate the accuracy. The role of the optimiser in mitigating the loss will be discussed in the Results and Discussion section. During the training process, the validation accuracy was also computed upon completion of the training process of each epoch. An early stopping mechanism was utilised to stop the training process if accuracy was found to have degraded.

6. Results and Discussion

In this section, we deliberate on the results of our set of experiments conducted on two Roman Urdu datasets as well as the IMDB Movie reviews dataset for sentiment analysis. The results as tabulated in Table 4 determines the validation accuracy of the RUECD dataset. It can be clearly observed that binary classification outperforms the ternary classification task, as depicted in Figure 5. Our proposed model clearly outperforms the baseline deep learning models under three different neural word embedding schemes.

6.1. Sentiment Analysis by RU-BiLSTM Model

We performed the sentiment analysis task using our attention-based deep learning model for both binary and ternary classification. The results of our RU-BiLSTM model in Table 4 show that RUECD performs better for binary classification with an accuracy of 80.63% compared with RUSA-19, which triggers a slightly lower accuracy of 77.50%. Moreover, in the case of ternary classification (where the classifier has to classify an additional neutral class along with the positive and negative ones), the accuracy achieved was 67%. However, it can be observed from Figures 5 and 7 that the model gained accuracy on earlier epochs and then gradually declined in accuracy, which is basically due to the overfitting problem. To overcome this, we employed the early-stopping mechanism, which stops the learning process of the parameters. On the other hand, the incurred loss for validation data is high compared with the training data, as illustrated in Figure 8. The increase in loss score is due to a misrepresentation of one class, which is more penalised by the cross entropy loss function. However, the incurred loss has no direct relation with accuracy; rather, its a log likelihood function that estimates the error rate, as discussed earlier. Analysing the class-wise evaluation as recorded in Table 5 reveals that the positive class achieved better accuracy compare with the negative and neutral classes.

Table 4. Validation accuracy of our RU-BiLSTM model.

Dataset	Ternary (%)	Binary (%)
RUECD	67.00	80.63
RUSA-19	65.15	77.50

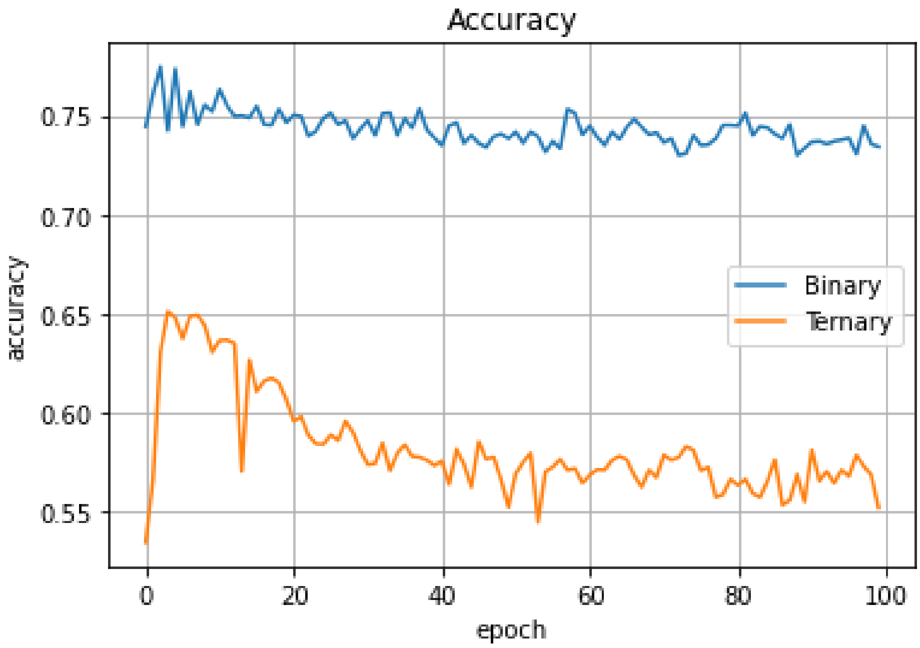


Figure 7. Validation accuracy comparison of binary and ternary classification on the RUSA-19 Dataset. The graph clearly depicts that binary classification outperformed the ternary classification.

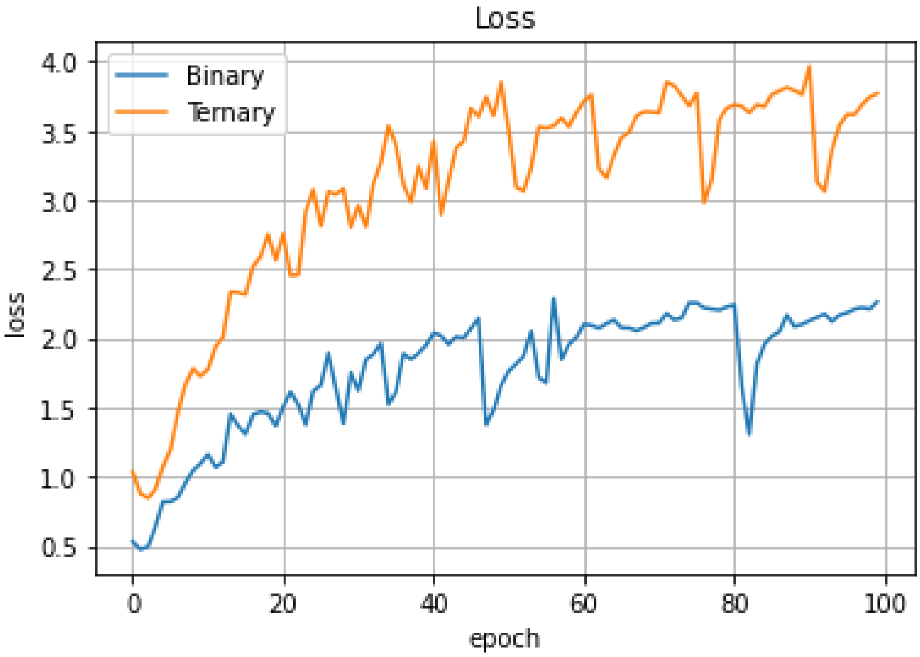


Figure 8. Loss comparison of the binary and ternary classifications.

Table 5. Class-wise precision and recall chart for the RUECD dataset.

Class	Precision	Recall	F1 Score
Negative (0)	0.66	0.65	0.65
Positive (1)	0.73	0.74	0.74
Neutral (2)	0.64	0.63	0.63
Average	0.68	0.67	0.67
Accuracy			0.67

6.2. Comparison with Baseline Methods

This section is intended to compare our model with baseline models so as to evaluate the model and to conclude the results. The results of baseline models among with our model are tabulated in Table 6.

Table 6. Comparison of different evaluation measures of baseline models with our proposed model for Roman Urdu Sentiment Analysis.

Model	Word Embedding	Evaluation Metric			
		PRE	REC	F1	ACC
SVM [13]	TFIDF	0.60	0.60	0.60	0.60
CNN [57]	Word2vec	0.62	0.59	0.57	0.58
	FastText	0.60	0.60	0.60	0.60
	Glove	0.61	0.61	0.60	0.60
RNN	Word2vec	0.58	0.57	0.57	0.59
	FastText	0.53	0.51	0.50	0.51
	Glove	0.66	0.58	0.55	0.58
RCNN [12]	Word2vec	0.63	0.62	0.62	0.62
	FastText	0.62	0.61	0.60	0.61
	Glove	0.64	0.61	0.61	0.61
GRU [7]	Word2vec	0.63	0.62	0.62	0.62
	FastText	0.61	0.60	0.60	0.60
	Glove	0.61	0.60	0.60	0.60
LSTM [56]	Word2vec	0.62	0.62	0.62	0.62
	FastText	0.60	0.59	0.59	0.59
	Glove	0.61	0.61	0.61	0.61
BiLSTM (classic) [62]	Word2vec	0.64	0.64	0.64	0.64
	FastText	0.63	0.62	0.0.62	0.62
	Glove	0.63	0.61	0.61	0.61
RU-BiLSTM (Our implementation)	Word2vec	0.68	0.67	0.67	0.67
	FastText	0.64	0.64	0.64	0.64
	Glove	0.65	0.65	0.65	0.65

The performance of our proposed method among with SOTA methods of deep learning is reported in Table 6. The efficacy of all models is evaluated using three famous feature representation approaches: word2vec [18], Glove [59], and FastText [60]. The evaluation metrics used are precision, recall, F1-score, and accuracy. It can be observed that CNN [57] has shown average accuracies of 0.58 and 0.60 under three different feature representation schemes. However, the precision of 0.62 achieved in the case of word2vec beats many other deep networks. The classic RNN has also shown a performance on par with that of CNN in terms of accuracy; however, the precision score of 0.66 gained under the umbrella of Glove features outperforms all other methods except our proposed model. It is worth mentioning here that classic RNN has shown a poor performance for the FastText feature representation, with an accuracy of 0.51, whereas CNN achieved a better accuracy of 0.60 under the same representation. The RCNN [12] achieved a better score in all three feature representation schemes, with a maximum accuracy of 0.62. On the other hand, GRU [7], which was recently proposed for Roman Urdu sentiment analysis, also showed average accuracies of 0.62 and 0.60. The classic LSTM [56] has also shown a performance equivalent to CNN, RCNN, and GRU, which is against the expectation. Accuracies of 0.62 (word2vec), 0.59 (FastText), and 0.61 (Glove) were achieved by LSTM. The Bidirectional LSTM (BiLSTM), which was recently proposed for emotion detection [62], has shown a slight improvement and scored an accuracy of 0.64 under the umbrella of word2vec. Reasonable scores of 0.62 and 0.61 were achieved on FastText and Glove, respectively. In order to mitigate this challenging task, where most of the SOTA methods failed to improve the performance, we propose an attention-based bidirectional network RU-BiLSTM. Our proposed method significantly improved the performance and outperformed the existing models. It can be observed that our proposed model has shown superior performance in all three embedding schemes and in all four evolutionary measures. However, under the word2vec embedding scheme, our proposed model showed the best performance, with a precision score of 0.68, a recall of 0.67, and 0.67 accuracy. In contrast, for Glove and FastText, accuracies of 0.65 and 0.64 were achieved, which is again far more better than existing models. In order to determine whether the proposed model is generalisable, we evaluated our model on the famous IMDB Movie Reviews dataset [63]. The classification results of our proposed model as reported in Table 7 are promising on this dataset with an accuracy of 0.88. On the other hand, the classic BiLSTM has also shown superior performance. The empirical evaluation reveals that the proposed attention-based model is not language-specific. However, it aimed at mitigating text that is informal in nature and needs to assign more weight to features with greater contribution.

Table 7. Sentiment analysis of IMDB Movie Reviews dataset.

Model	Word Embedding	Evaluation Metric			
		PRE	REC	F1	ACC
BiLSTM [62]	Word2vec	0.88	0.87	0.87	0.87
	FastText	0.87	0.86	0.86	0.86
	Glove	0.77	0.75	0.74	0.75
RUBiLSTM (Our implementation)	Word2vec	0.88	0.88	0.88	0.88
	FastText	0.87	0.86	0.86	0.86
	Glove	0.76	0.75	0.74	0.74

In a separate experiment on machine learning algorithms, we employed the TFIDF model with Support Vector Machine (SVM) [13] and experienced an accuracy not more than 61%.

6.3. Role of Optimisers

In our empirical evaluation, it was observed that optimisers that are used as backpropagation algorithm have a significant impact on loss. For the purpose of clarity, we used a variety of optimisation functions with versatile learning strategies by fine-tuning the hyperparameters. The Nadam, Adamax, and RMSprop optimisers have been found by mitigating loss. In contrast, with SGD, Adadelta and the classic adam optimiser have triggered a huge loss and slightly decreased the accuracy. Moreover, for SGD, the optimiser learning strategy has been found disastrous owing to the huge loss and lowest accuracy.

6.4. Discussion

Deep learning methodologies usually entail more trainable time, data, and fine-tuning of parameters compared with machine learning approaches. Machine learning schemes are much faster and require less time for training and typically use bag-of-words for feature engineering. The feature engineering that is generally adopted for machine learning classifiers suffers from the curse of dimensionality owing to a sparse matrix and the absence of word vectors. In contrast, deep networks are found to be much more reliable in feature engineering as well as in extracting hidden patterns. The deep neural network ability of pattern recognition enables us to semantically empower the model. Furthermore, such networks, if powered by neural word embedding, could yield promising results.

We also employed three neural word embedding schemes: word2vec, Glove, and FastText that were trained on an ample amount of Roman Urdu text. In comparison with pre-trained word embedding, the random word embedding entails more parameters to train and attains less classification accuracy [64]. In sentiment analysis task, feature extraction and the design of classifier are two significant factors. The LSTM has shown excellent performance in many NLP problems; however, the performance can still be ameliorated. We proposed an attention based bidirectional LSTM-based deep neural network RU-BiLSTM for Roman Urdu sentiment analysis.

In our series of experiments for Roman Urdu sentiment analysis, we observed various factors affecting the classification accuracy. It is observed that the attention layer has a significant effect on ameliorating the classification accuracy owing to concentrating on what is more important. On the other hand, the bidirectional LSTM's capability of learning the network in both directions is vivid in the results. It is quite evident that traditional RNN lacks the capability of handling long-term temporal dependencies, which restricts the classification accuracy of this model. By fine-tuning the embedding dimensions, it is observed that word embedding size also plays a vital role in establishing the semantic relationship between the features and, consequently, improves the performance. Moreover, among three neural word embedding, word2vec has shown superior performance in comparison with Glove and FastText.

We also performed experiments by comparing words' weights before and after the attention layer. It is observed that the attention layer significantly modifies weights of many words, especially words that are subjective, i.e., conveying sentiment in Urdu. That is likely the reason why adding an attention layer to BiLSTM improves the performance of the model. Experiments and careful analysis also show that the accuracy of the proposed model could not be further improved because of the main limitations that are discussed in the Introduction section, i.e., several variations in spelling for the same words and multilinguality since Roman Urdu is very informal in writing and people use different spelling for the same words. Sometimes, people express their feelings in words and change the spelling unnecessarily. For example, for the sentence "really enjoyed (in English)", in Roman Urdu, it is written as "boot maza aya". This same sentence can be written by some people as boooooooot maaaaaazaaaaa ayaaaaaaa". However, this can be handled by a robust stemmer, which is lacking for Roman Urdu. We used the stemmer of our previous work [55], which has an impact on improving the accuracy. However, more attention on robust stemmers is needed.

In our experimental trials, it is discovered that pre-trained word embedding outperforms the randomly initialised word embedding with an almost 4% improvement. Our proposed RU-BiLSTM model uses bidirectional LSTM, which has the capability to intelligently comprehend both preceding and succeeding contexts. We tried to evaluate both short and long user reviews, which were beautifully dealt with by our bidirectional network. Additionally, we employed the attention layer to further improve the performance of our model. The primary purpose of the attention mechanism is to determine the impact of each word on the phrase. It can capture the key components of sentence semantics by assigning attention weights to each word. The combination of these approaches enhances RU-BiLSTM's classification abilities and enhances its grasp of sentence semantics.

In contrast, the convolutional neural network-based approaches, such as static CNN, which uses local correlation for temporal structure of spatial data from anywhere in the text, can extract the n-gram features using convolutional filters. Our experimental results shown in Table 6 determine that CNN's approach to dealing with text features is not very reliable owing to an arbitrary sequence of feature selection that restricts our understanding of sentence semantics. An accuracy of 0.60 is achieved for CNN-based approach. Likewise, the RCNN [12] model, which uses both CNN and RNN layers, achieved accuracies of 0.62 under word2vec embeddings and 0.61 using Glove and FastText. This is possibly because of the CNN layer, which reduces the parameters and, consequently, downgrades the performance. The GRU, which was recently proposed for Roman Urdu sentiment analysis, has also shown a performance on par with those of CNN and RCNN. In general, GRU and LSTM are perceived as similar networks. However, owing to the lower number of gates in GRU, the LSTM is preferred (https://towardsdatascience.com/grus-and-lstm-s-741709a9b9b1) (accesed on 14 February 2022). The experiment on standalone LSTM depicts that having the capability to memorise the preceding context is insufficient for comprehending the sentence semantics. Rather, a bidirectional approach is needed to mitigate this problem, as proposed in our model.

We also evaluated the classic RNN, which suffers from a short-term memory problem. The classic RNN fails to capture the contextual information from earlier time-stamps in case the length of sequence is long. The problem is known as the gradient vanishing problem, which LSTM can better deal with. However, the classic LSTM performance is not found satisfactory, with an accuracy of 0.62. On the other hand, our proposed model RU-BiLSTM, as discussed above, can better preserve the contextual information in the Roman Urdu case too. Since our dataset contains both short and long reviews, they are evaluated on our proposed model and attained superior performance compared with baseline models. An improvement of almost 8% is achieved on the RU-BiLSTM model. Hence, our proposed model outperforms the baseline models.

Beside some astounding features of SVM coupled with the TFIDF model, the sentiment analysis results for Roman Urdu in multi-class environment was not very satisfactory. Even if the SVM has the advantage of tuning fewer parameters than deep networks, it is still quite very tricky to choose the right kernel and other parameters such as the width of the hyperplane.

The empirical evaluation also reveals that, on a higher number of epochs, the models start increasing in loss and consequently the accuracy suffers. In order to avoid overfitting, the model was trained on a smaller number of epochs and a low learning rate was set to learn the optimal weights. It was also observed that, if we increase the dropout rate to regularise the weights, it gravely affects the accuracy. However, a dropout rate of 0.2 to 0.5 has a tiny effect on the performance of the model.

7. Conclusions

In E-commerce, a sentiment analysis task on text, which has a colloquial nature, has several challenges. Among these challenges, feature extraction and design of classifiers are the significant ones. Another ability that semantically empowers the deep learning model is pattern recognition. Most of the baseline deep neural networks lack this capability. The classic LSTM has solved this problem to some extent, but it still needs improvement. This research study proposed an attention-based bidirectional LSTM network, RU-BiLSTM, that has shown superior performance on our newly generated Roman Urdu dataset, RUECD. The dataset is one of the major contributions of this paper, and it is openly available (https://github.com/bilalbaloch1/RU-BiSLTM/blob/main/RUECD.csv) (accesed on 15 February 2022) along with all the model code. The code and dataset have been made public so that future researchers may improve upon the accuracy of the dataset. Roman Urdu being a resource-starved language has always needed a benchmark dataset. This study presented the largest ever Roman Urdu dataset containing 26K+ user reviews carefully annotated by three annotators with excellent knowledge of Roman Urdu. The dataset and API for Roman Urdu sentiment analysis is publicly available for future research in this area. We conducted a series of experiments in the presence of three different neural word embedding for Roman Urdu: word2vec, Glove, and FastText. Among three word embedding schemes, word2vec embeddings has shown better performance. The neural word embedding and attention layer are used to semantically empower the model and to enhance the pattern extraction capability. The experimental results reveals that the classic LSTM, RNN, RCNN, and CNN have shown an average level performance. On the other hand, our proposed model RU-BiLSTM has shown significant improvement in all four evolutionary metrics i.e., precision, recall, F1-measure, and accuracy. The experimental results shows that our proposed model outperforms the baseline models, with enhanced semantic capability and a significant improvement of 8% in terms of accuracy.

Future work regarding Roman Urdu focuses on the creation of a benchmark dataset for fine-grained sentiment analysis with research in transformer learning and on designing a new self-attention mechanism to further improve the performance. Future research is based on the following agendas:

1. The development of a benchmark Roman Urdu dataset to support fine-grained sentiment analysis and question classification. The fine-grained sentiment analysis will include five types of discrete classifications namely, strongly negative, weakly negative, neutral, weakly positive, and strongly positive.
2. Designing an new improved self-attention mechanism to further ameliorate the performance.
3. Employment of sentence transformer learning such as BERT to estimate its performance for Roman Urdu NLP tasks.

Author Contributions: Conceptualisation, B.A.C., J.B. and M.B.; methodology, B.A.C., J.B., M.B., S.M.D. and A.S.I.; validation, B.A.C., A.S.I. and S.M.D.; formal analysis, B.A.C. and J.B.; investigation, B.A.C., J.B. and S.M.D.; resources, J.B. and A.S.I.; data curation, B.A.C. and J.B.; writing—original draft preparation, B.A.C., J.B. and M.B.; writing—review and editing, A.S.I. and S.M.D.; visualisation, B.A.C. and M.B.; supervision, J.B. and A.S.I.; project administration, J.B. and A.S.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BPTT	Back-Propagation Through Time
NLP	Natural Language Processing
RNN	Recurrent Neural Network
GRU	Gradient Recurrent Unit
LSTM	Long Short-Term Memory
BiLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
RU	Roman Urdu
RUSA-19	Roman Urdu Sentiment Analysis Dataset
RUECD	Roman Urdu Dataset
SVM	Support Vector Machine
API	Application Programming Interface
SNN	Sandwich Neural Network
S2SAN	Sequence-to-Sequence Attention Network
SLR	Systematic Literature Review
DFST	Discriminative Feature Spamming Method
TUC	Term Utility Criteria

References

1. Ligthart, A.; Catal, C.; Tekinerdogan, B. Systematic reviews in sentiment analysis: A tertiary study. *Artif. Intell. Rev.* **2021**, *54*, 4997–5053. [CrossRef]
2. Imran, A.S.; Daudpota, S.M.; Kastrati, Z.; Batra, R. Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on COVID-19 related tweets. *IEEE Access* **2020**, *8*, 181074–181090. [CrossRef]
3. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26. [CrossRef]
4. Birjali, M.; Kasri, M.; Beni-Hssane, A. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowl. Based Syst.* **2021**, *226*, 107134. [CrossRef]
5. Kastrati, Z.; Dalipi, F.; Imran, A.S.; Pireva Nuci, K.; Wani, M.A. Sentiment Analysis of Students' Feedback with NLP and Deep Learning: A Systematic Mapping Study. *Appl. Sci.* **2021**, *11*, 3986. [CrossRef]
6. Safdar, Z.; Bajwa, R.S.; Hussain, S.; Abdullah, H.B.; Safdar, K.; Draz, U. The role of Roman Urdu in multilingual information retrieval: A regional study. *J. Acad. Librariansh.* **2020**, *46*, 102258. [CrossRef]

7. 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]
8. Feldman, R. Techniques and Applications for Sentiment Analysis. *Commun. ACM* **2013**, *56*, 82–89. [CrossRef]
9. D'Andrea, A.; Ferri, F.; Grifoni, P.; Guzzo, T. Approaches, Tools and Applications for Sentiment Analysis Implementation. *Int. J. Comput. Appl.* **2015**, *125*, 26–33. [CrossRef]
10. Bakshi, R.K.; Kaur, N.; Kaur, R.; Kaur, G. Opinion mining and sentiment analysis. In Proceedings of the 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 16–18 March 2016; pp. 452–455.
11. Dias, L.; Gerlach, M.; Scharloth, J.; Altmann, E.G. Using text analysis to quantify the similarity and evolution of scientific disciplines. *R. Soc. Open Sci.* **2018**, *5*, 171545. [CrossRef]
12. 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]
13. Noor, F.; Bakhtyar, M.; Baber, J. Sentiment analysis in E-commerce using SVM on roman urdu text. In Proceedings of the International Conference for Emerging Technologies in Computing, London, UK, 19–20 August 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 213–222.
14. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef] [PubMed]
15. Zhou, C.; Sun, C.; Liu, Z.; Lau, F. A C-LSTM neural network for text classification. *arXiv* **2015**, arXiv:1511.08630.
16. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882.
17. Alam, M.; Samad, M.D.; Vidyaratne, L.; Glandon, A.; Iftekharuddin, K.M. Survey on deep neural networks in speech and vision systems. *Neurocomputing* **2020**, *417*, 302–321. [CrossRef]
18. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2013; pp. 3111–3119.
19. Rao, G.; Huang, W.; Feng, Z.; Cong, Q. LSTM with sentence representations for document-level sentiment classification. *Neurocomputing* **2018**, *308*, 49–57. [CrossRef]
20. Hemmatian, F.; Sohrabi, M.K. A survey on classification techniques for opinion mining and sentiment analysis. *Artif. Intell. Rev.* **2019**, *52*, 1495–1545. [CrossRef]
21. Yadav, A.; Vishwakarma, D.K. Sentiment analysis using deep learning architectures: A review. *Artif. Intell. Rev.* **2020**, *53*, 4335–4385. [CrossRef]
22. Ghafoor, A.; Imran, A.S.; Daudpota, S.M.; Kastrati, Z.; Batra, R.; Wani, M.A. The Impact of Translating Resource-Rich Datasets to Low-Resource Languages Through Multi-Lingual Text Processing. *IEEE Access* **2021**, *9*, 124478–124490. [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. Rani, S.; Kumar, P. Deep learning based sentiment analysis using convolution neural network. *Arab. J. Sci. Eng.* **2019**, *44*, 3305–3314. [CrossRef]
25. Yue, L.; Chen, W.; Li, X.; Zuo, W.; Yin, M. A survey of sentiment analysis in social media. *Knowl. Inf. Syst.* **2019**, *60*, 617–663. [CrossRef]
26. Garcia, K.; Berton, L. Topic detection and sentiment analysis in Twitter content related to COVID-19 from Brazil and the USA. *Appl. Soft Comput.* **2021**, *101*, 107057. [CrossRef] [PubMed]
27. Nassif, A.B.; Elnagar, A.; Shahin, I.; Henno, S. Deep learning for Arabic subjective sentiment analysis: Challenges and research opportunities. *Appl. Soft Comput.* **2020**, *98*, 106836. [CrossRef]
28. 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]
29. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2222–2232. [CrossRef]
30. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent Convolutional Neural Networks for Text Classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, Austin, TX, USA, 25–30 January 2015; AAAI Press: Palo Alto, CA, USA, 2015; pp. 2267–2273.
31. Zhan, Z.; Hou, Z.; Yang, Q.; Zhao, J.; Zhang, Y.; Hu, C. Knowledge attention sandwich neural network for text classification. *Neurocomputing* **2020**, *406*, 1–11. [CrossRef]
32. Haque, T.U.; Saber, N.N.; Shah, F.M. Sentiment analysis on large scale Amazon product reviews. In Proceedings of the International Conference on Innovative Research and Development (ICIRD), Bangkok, Thailand, 11–12 May 2018; pp. 1–6.
33. Rathor, A.S.; Agarwal, A.; Dimri, P. Comparative Study of Machine Learning Approaches for Amazon Reviews. *Procedia Comput. Sci.* **2018**, *132*, 1552–1561. [CrossRef]
34. Zhang, Y.; Wang, J.; Zhang, X. Personalized sentiment classification of customer reviews via an interactive attributes attention model. *Knowl. Based Syst.* **2021**, *226*, 107135. [CrossRef]
35. Wang, P.; Li, J.; Hou, J. S2SAN: A sentence-to-sentence attention network for sentiment analysis of online reviews. *Decis. Support Syst.* **2021**, *149*, 113603. [CrossRef]
36. Elfaik, H.; Nfaoui, E.H. Deep bidirectional lstm network learning-based sentiment analysis for arabic text. *J. Intell. Syst.* **2021**, *30*, 395–412. [CrossRef]

37. Gan, C.; Feng, Q.; Zhang, Z. Scalable multi-channel dilated CNN–BiLSTM model with attention mechanism for Chinese textual sentiment analysis. *Future Gener. Comput. Syst.* **2021**, *118*, 297–309. [CrossRef]
38. Fares, M.; Moufarrej, A.; Jreij, E.; Tekli, J.; Grosky, W. Unsupervised word-level affect analysis and propagation in a lexical knowledge graph. *Knowl. Based Syst.* **2019**, *165*, 432–459. [CrossRef]
39. Cheng, Y.; Gong, Y.; Liu, Y.; Song, B.; Zou, Q. Molecular design in drug discovery: A comprehensive review of deep generative models. *Brief. Bioinform.* **2021**, *22*, bbab344. [CrossRef]
40. Abboud, R.; Tekli, J. Integration of nonparametric fuzzy classification with an evolutionary-developmental framework to perform music sentiment-based analysis and composition. *Soft Comput.* **2020**, *24*, 9875–9925. [CrossRef]
41. Duan, J.; Luo, B.; Zeng, J. Semi-supervised learning with generative model for sentiment classification of stock messages. *Expert Syst. Appl.* **2020**, *158*, 113540. [CrossRef]
42. Sun, Y.; Kamel, M.S.; Wong, A.K.; Wang, Y. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognit.* **2007**, *40*, 3358–3378. [CrossRef]
43. Srividya, K.; Mary Sowjanya, A. NA-DLSTM—A neural attention based model for context aware Aspect-based sentiment analysis. *Mater. Today Proc.* **2021**, *388*, 135–143. [CrossRef]
44. de Haro-García, A.; Cerruela-García, G.; García-Pedrajas, N. Ensembles of feature selectors for dealing with class-imbalanced datasets: A proposal and comparative study. *Inf. Sci.* **2020**, *540*, 89–116. [CrossRef]
45. Niu, K.; Zhang, Z.; Liu, Y.; Li, R. Resampling ensemble model based on data distribution for imbalanced credit risk evaluation in P2P lending. *Inf. Sci.* **2020**, *536*, 120–134. [CrossRef]
46. He, H.; Garcia, E.A. Learning from Imbalanced Data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284.
47. Sun, Y.; Wong, A.K.; Kamel, M.S. Classification of imbalanced data: A review. *Int. J. Pattern Recognit. Artif. Intell.* **2009**, *23*, 687–719. [CrossRef]
48. Shaikh, S.; Daudpota, S.M.; Imran, A.S.; Kastrati, Z. Towards Improved Classification Accuracy on Highly Imbalanced Text Dataset Using Deep Neural Language Models. *Appl. Sci.* **2021**, *11*, 869. [CrossRef]
49. Khan, N.S.; Sehar, S.; Butt, A.H. A Systematic Literature Review on Urdu Sentiment Analysis. *Int. J. Disaster Recovery Bus. Contin.* **2021**, *12*, 550–575.
50. Mehmood, K.; Essam, D.; Shafi, K.; Malik, M. An unsupervised lexical normalization for Roman Hindi and Urdu sentiment analysis. *Inf. Process. Manag.* **2020**, *57*, 102368. [CrossRef]
51. 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]
52. 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]
53. Arif, H.; Munir, K.; Danyal, A.S.; Salman, A.; Fraz, M.M. Sentiment analysis of roman urdu/hindi using supervised methods. *Proc. ICICC* **2016**, *8*, 48–53.
54. Naqvi, R.A.; Khan, M.A.; Malik, N.; Saqib, S.; Alyas, T.; Hussain, D. Roman Urdu news headline classification empowered with machine learning. *Comput. Mater. Contin.* **2020**, *65*, 1221–1236.
55. Chandio, B.; Shaikh, A.; Bakhtyar, M.; Alrizq, M.; Baber, J.; Sulaiman, A.; Rajab, A.; Noor, W. Sentiment Analysis of Roman Urdu on E-Commerce Reviews Using Machine Learning. *CMES-Comput. Model. Eng. Sci.* **2022**. [CrossRef]
56. 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]
57. Rizwan, H.; Shakeel, M.H.; Karim, A. Hate-speech and offensive language detection in roman Urdu. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 2512–2522.
58. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
59. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
60. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2016**, *5*, 135–146. [CrossRef]
61. Raffel, C.; Ellis, D.P. Feed-forward networks with attention can solve some long-term memory problems. *arXiv* **2015**, arXiv:1512.08756.
62. Joshi, V.M.; Ghongade, R.B.; Joshi, A.M.; Kulkarni, R.V. Deep BiLSTM neural network model for emotion detection using cross-dataset approach. *Biomed. Signal Process. Control* **2022**, *73*, 103407. [CrossRef]
63. Maas, A.L.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning Word Vectors for Sentiment Analysis. In *Human Language Technologies, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, OR, USA, 19–24 June 2011*; Association for Computational Linguistics: Portland, OR, USA, 2011; pp. 142–150.
64. Liu, G.; Guo, J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing* **2019**, *337*, 325–338. [CrossRef]