Lexical Variation and Sentiment Analysis of Roman Urdu Sentences with Deep Neural Networks

Muhammad Arslan Manzoor[1], Saqib Mamoon[2], Song Kei Tao[3], Ali Zakir[4], Muhammad Adil[5], Jianfeng Lu*[6]
School of Computer Science and Engineering
Nanjing University of Science and Technology
Nanjing, China

***Abstract*—Sentiment analysis is the computational study of reviews, emotions, and sentiments expressed in the text. In the past several years, sentimental analysis has attracted many concerns from industry and academia. Deep neural networks have achieved significant results in sentiment analysis. Current methods mainly focus on the English language, but for minority languages, such as Roman Urdu that has more complex syntax and numerous lexical variations, few research is carried out on it. In this paper, for sentiment analysis of Roman Urdu, the novel "Self-attention Bidirectional LSTM (SA-BiLSTM)" network is proposed to deal with the sentence structure and inconsistent manner of text representation. This network addresses the limitation of the unidirectional nature of the conventional architecture. In SA-BiLSTM, Self-Attention takes charge of the complex formation by correlating the whole sentence, and BiLSTM extracts context representations to tackle the lexical variation of attended embedding in preceding and succeeding directions. Besides, to measure and compare the performance of SA-BiLSTM model, we preprocessed and normalized the Roman Urdu sentences. Due to the efficient design of SA-BiLSTM, it can use fewer computation resources and yield a high accuracy of 68.4% and 69.3% on preprocessed and normalized datasets, respectively, which indicate that SA-BiLSTM can achieve better efficiency as compared with other state-of-the-art deep architectures.**

***Keywords*—Sentiment analysis; Self-Attention Bidirectional LSTM (SA-BiLSTM); Roman Urdu language; review classification**

I. Introduction

Sentiment analysis is a fundamental task that classifies the feedback, feelings, emotions, and gestures in natural language processing domain [1]. Recent theoretical developments have revealed that every discussion on social media, forums, blogs, chats has a great influence on society regardless of the region or the language. This situation is considerable for vast number of societies and business communities in terms of feedback, to conquer deficiencies and enhance productivity. The growing demand for the computational learning of text, further results in sentence classification, aspect categorization, and opinion detection.

Convolutional Neural Networks (CNNs) have achieved impressive results on the important task of sentence categorization [2]. Further, Recurrent Neural Networks (RNNs) and their variants such as LSTM, BiLSTM, and GRU have produced better results for sequence and language modelling [3], [4]. Previous studies show that most of the neural networks require more time and memory resources to train and run the model and difficult to optimize. A solution to this problem is proposed by Bahdanau et al. [5] which emphasized that Attention keeps track of the source input sequence by building a shortcut between encoder hidden states and context vector. This study gave a great break through at the Language Modeling Planet by introducing Transformer [6] that is merely based on Attention mechanism, drops off recurrence and convolutions thoroughly in terms of training and striking results. In terms of training performance, Attention mechanism is more stable as there are no large number of hidden states to update and maintain. After it, Attention networks have been applied to multiple tasks [7]–[9] i.e., image classification, text summarizer , and sentiment analysis.

Another key limitation is that most of these models are unidirectional. In order to address this issue, a novel framework "The Self-Attention Bidirectional LSTM (SA-BiLSTM)" is proposed in this study. In SA-BiLSTM, Self-Attention mechanism focuses only the relevant word embedding to correlate in the whole sentence which influences polarity and BiLSTM supervise context representations of these attended embedding in forward and backward direction. Studies mentioned above are evidenced that Self-Attention can produce better result and consume less resources because of its selective nature and Bidirectional LSTM is integrated to conquer the limitation of unidirectional model.

A lot of research work has been done on English language Analysis [10]. According to our best knowledge, no previous research is carried out to classify the sentences of subcontinent language (Urdu/Hindi) with Neural Networks. Urdu is the native language of Pakistan and currently being spoken and understood in several parts of India, Bangladesh, and Nepal [11]. Roman Urdu is one of those languages which is usually used on social media for communication and comments [12]. There is no dataset of Roman Urdu available that is ready to apply deep learning models. The most challenging task was to preprocess and normalize the Roman Urdu dataset then made it usable for Sentiment Analysis.

In view of the existing gap, our contributions is as follows:

- Preprocessed the 10,000 Roman Urdu Sentences (negative and positive reviews), and normalized more than 3000 sentences.

*Corresponding author

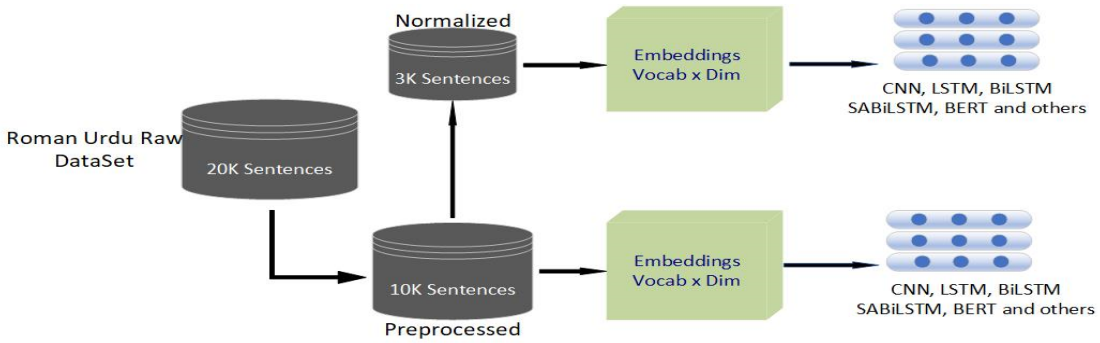


Fig. 1. Dataflow diagram from preprocessing of dataset to implementing the models

- Bidirectional LSTM is integrated with Self-Attention to deal with the complexity of sentences and variations of words in Roman Urdu.
- Self-Attention with Bidirectional LSTM (SA-BiLSTM) is trained and evaluated on Roman Urdu sentences, then comparison and analysis are made with other models.

This study has made a significant contribution by addressing the problems of Roman Urdu scripts with deep neural networks. The layout of this paper is as follows. Initially, section II describes the existing networks for NLP tasks. Section III proposes the language architecture in methodology. Then, the Section IV reveals the dataset preprocessing and experiments. Moreover, Section V illustrates the results and analysis. Finally, Section VI concludes the research work and opens the future.

II. Related Work

In recent years, deep learning has become the key technique for many researchers to deal with sentiment analysis. It consists of effective models that are used to solve a variety of problems efficiently [13]. Convolution Neural Network (CNN) has achieved impressive results on the task of sentence categorization [2], [14]–[17]. The models need not be complex to realize strong results [2], regarding visual sentiment analysis, CNN enhanced its efficiency by growing its size and depth [18]. Results show that the proposed system achieves high performance without fine-tuning. Detailed research [19] using Convolutional Neural Network (CNN) has presented a summary of sentiment analysis related to micro-blog. However, in terms of sentence processing, CNN extracts the feature without correlating all the sentence that leads to producing low results and consume high resources.

Recurrent Neural Network (RNN) is well known for sequential information processing as they use internal memory states to process input sequences [20]. It produces output that is dependent on the computation of all previous input and hidden states. RNNs prefer terms that they get later in the sentence, despite the words they get earlier. RNN lacks in most applications because they demand high memory, time and hardware resources.

To deal with the shortcoming of standard RNN, researchers have developed sophisticated variants of RNN [21]. Bidirectional RNN is built on the idea that the outcome at each time may not only bases on the previous elements but also depends on the next elements in the sequence. LSTM cell uses forget gate, input gate and output gate for processing cell states to focus most concerning information which enhances the performance of this cell [22]. Gated Recurrent Unit (GRU) combined the forget gate and input gate to make it simpler but less efficient than LSTM for long sequences and large datasets [3]. BiLSTM (Bidirectional long short-term memory) is an extended version of LSTM with more information [23]. BiLSTM access both the preceding and succeeding contexts by considering the forward and the backward hidden layers employed by Chen et al. [24] for sentiment analysis task. All these variants of RNNs have achieved great success in numerous tasks. However, they are often called as black boxes, lacking interpretability and consume high resources [25]. Research efforts to solve this issue have steadily increased.

The Attention mechanism was presented to upgrade the RNN encoder decoder sequence-to-sequence network for NMT [5], [26]. Initially, Attention was defined as the process of determining a context vector for the next decoder step that consists of the most relevant information with the encoder hidden states. Seminal contributions have been made by Vaswani et al. [6] when Transformer architecture was proposed for machine translation. It depends only on Attention mechanisms, as the best replacement of either recurrent or convolution neural networks. For sequence processing and language modeling, Transformer has outperformed the recurrent neural network and their variants.

A closer look to the literature on neural networks for sentence classification [5], [6] reveal that Attention predicts based on only recent hidden states (unlike RNN, which predict based on entire history and reminds all the previous hidden states). The objective is to devise and implement a system that consists of Self-Attention to address the problem of complex structure of Roman Urdu Sentences. In this study, a more efficient and lightweight model Self-Attention Bidirectional LSTM is proposed for targeted problem, where Self-Attention takes charge of the complex formation by correlating the whole sentence and determining embedding that consists of the most relevant information. Bidirectional LSTM is integrated to strengthen the network as it extracts

context representations to tackle the lexical variation of attended embedding in preceding and succeeding directions. Moreover, it promotes essential embedding by memorizing the contextual information for the long term. The results endorse that the integration of network leads to enhance the Self-Attention's performance. Besides, deficiencies of Bidirectional LSTM are conquered by Self-Attention module in the network.

III. Methodology

According to Bahdanau et al. [5], Attention's task is to compute the context vector for the succeeding decoder step that consists of maximum appropriate values of encoder hidden states after getting a weighted average of encoder hidden states. There was a factor of alignment score which represents the contribution to the weighted average between encoder states and previous decoder hidden states.

A. Self-Attention Mechanism

Following the above concept, Vaswani et al. [6] trained decoder hidden states as query vector which pay Attention to those hidden states of an encoder that have more influence in producing relevant output. Key, Value vectors are formed by hidden states of Encoder. Attention does not always take two different sentences and correlate them, it may take same sentence along column and row to extract the relation between different parts of it. Each sequence position is considered as Q and compared with the rest of sequence position K by correlating them and as a result V is produced that has most weighted relevance (Self-Attention). Initially, compatibility function determines the weights connecting the query and the keys in (1). Compatibility score is transformed by the softmax function into probability distribution as described in (2), this normalization helps Query (q) to consider the important tokens Key (k) for classification. Then, weighted average of Value (v) vectors corresponding (k) produced output. Feed forward layers and learned linear projections were applied to create (query, value, key) vectors. Taking a query q, values and keys, compatibility function is responsible to compute correlating outcome between k and q as follows

$$f(k,q) = \frac{(k)(q)^T}{\sqrt{d_k}} \tag{1}$$

d_k is served as a scaling operator, and maintains the numerical stability when the dimension of keys increases. The softmax function is applied to the compatibility score to compute Weighted sum α.

$$a = softmax\left\{f(k,q)\right\} \tag{2}$$

$$Z = \sum a(v) \tag{3}$$

Equation (3) represents the most relevant values with the query selected by the highest weights.

B. Bidirectional LSTM

Attention output is passed to Bidirectional LSTM to memorize only the most considerable Self-Attended preceding and succeeding embedding which would enhance the accuracy. LSTM support to reminisce those embeddings by means of its three gates architecture to impact the results efficiently. To strengthen the LSTM and deal the weakness (not accessing the forward hidden layers for the future token), Bidirectional LSTM is used to collect the contextual and relevance information from previous and future embedding values.

C. Positional Information

Input embeddings gather the positional Information of sequence ordering through the Position Embedding Layer. Absolute (or relative) positional information of each token in a sequence is passed to Attention layer. A method is proposed where positional encoding (PE) vectors are formed using sine and cosine functions of difference frequencies and then are appended to the input embeddings [6].

D. Network Architecture (SA-BiLSTM)

The Network architecture of SA-BiLSTM is represented in Fig. 2. The input sentences are passed through the embedding layer that uses pre-trained embedding generated from Word2Vec model. Embeddings are passed to Position Encoding (PE) to learn position representation as Attention receive the whole sequence and does not keep the positional information. The output embeddings of PE are sent to SA-BiLSTM module which is described below:

Self-Attention mechanism is applied to every position of source sentence. For each sentence position query, key, and value would behave as vectors. The absence of previous decoder state that behave as query made every position of input sequence as a set of Query vectors. In this step, by Keeping each query static, compatibility score with all the rest of keys of that sequence would be measured. It is applied to all values of vectors $O = (o_1, o_2, ..o_n)$ which creates Output vector which has the information of how much each sequence query is relative to the rest of queries and contributes in polarity of sentence.

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{4}$$

It led to multi-head Attention which implements parallel computing on the whole sequence by making groups of the query, keys, and values in Q, K, and V matrices respectively. After the above study, we propose a Self-Attention Bi-LSTM Sequential Model (SA-BiLSTM) for the normalized dataset which consists of three major sections. Input Sentences are converted into Embedding with Q, K and V vectors which are passed through the Position Embedding module that brings consideration to sequence ordering. These Embedding with positional Information is passed to Self-Attention Module which applies Attention mechanism as in (4) on each sequence position. It helps in a correlating the weights. Multi head Attention performed Attention h times on (Q, K, V) matrices of dimension d_{model}/h in (5). where each head performed Self-Attention to produce an output of dimension d_{model}/h in (6). Then the outputs are concatenated to produce matrices of

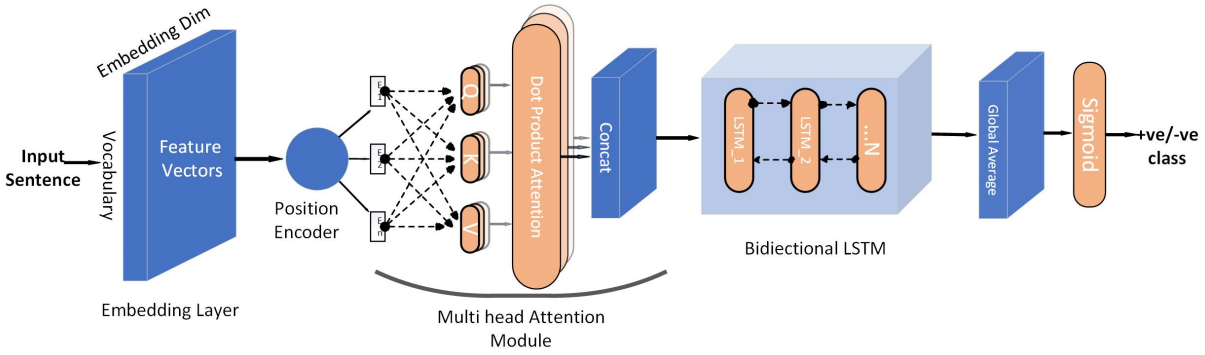


Fig. 2. Network Architecture of SA-BiLSTM

identical dimensionality to Self-Attention on the actual (Q, K, V) matrices. Feed forward layers pass the embedding to next module.

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W \quad (5)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (6)$$

These Self-Attended Embedding sent to stack BiLSTM for contextual semantic information on the backward and forward direction of embeddings. It selects those embedding which is going to influence polarity more by memorizing its previous effect in both directions.

$$\tilde{c} = tanh(W_c[h_{t-1}, emb] + b_c) \quad (7)$$

$$c_t = f_t.c_{t-1} + i_t.\tilde{c_t} \quad (8)$$

$$h_t = o_t.tanh(c_t) \quad (9)$$

Equation (7) denotes the input to the cell, emb is the value selected by Self-Attention passed as input and controlled by hyperbolic tangent function, W_c and b_c are learnable parameters, ht-1 is the hidden state value of previous time step. In (8) and (9), i_t, $f_t$and o_t are the input, forget and output gate values activate by sigmoid function at time t, respectively. Cell state at current time step is denoted as c_t and c_{t-1} is cell state for previous time steps. The final output of cell at time t is filtered by the output gate denoted as h_t. Global average pooling is applied to the final output of the BiLSTM. Finally, sigmoid classifier can output the class of sentence.

IV. Experimental Setup

This section describes the experimental setup to analyze the performance of SA-BiLSTM for Roman Urdu sentiment analysis. All the well-known deep learning language models along with the proposed model are implemented on two datasets (preprocessed and normalized) of Roman Urdu sentences. Each sentence of the dataset is labeled with positive or negative tag 0,1. Experiments were run using a single Titan Pascal XP 12G. All models are implemented in Keras 2.2.4 with Tensorflow-GPU 1.13.1 backend using cuDNN 7.3.1 and CUDA 10.1.

A. Parameter Setting

Extensive experiments were run using adadelta, sgd, rmsprop, and adam optimizer. After ablation study, it is observed that adam optimizer using a 0.0004 learning rate with batch size of 32 achieved more stable results than rest of the optimizers. Cross-entropy loss with L2-regularization is performed on the model parameters with a λ value 10-3. The dropout value was kept 0.3 to avoid underfitting.The word embedding of 200 dimension created by word2vec model. For the activation purpose in final dense layer, sigmoid function is used.

B. Dataset

The dataset that is used to evaluate the deep learning models for Roman Urdu analysis is comprised of sentences extracted from Urdu blogs, social and news websites, prepared by Sharf et al. [27], where reviews written by customers, such as social media users and fan followers of celebrities. The dataset available at resource* which contains more than 20,000 sentences (positive, negative and neutral) that belongs to 4 to 5 domains of online platforms.

TABLE I. Sample Sentences of Roman Urdu from Dataset

Roman Urdu	English
usay saalgira per **khoobsurat** tohfa mila	He got a beautiful birthday present
wo aik **kamyaab** shakhs hai	He is a successful person
us movie ka subject bohat **acha** hai	The theme of this movie is very good
isay **bura** samjha jata hai	This is considered bad

*1) **Preprocessed dataset**:* We mainly focused on binary (positive, negative) classification and selected 10,000 most appropriate sentences from resource for preprocessing and termed it as "Preprocessed" dataset in experiment. To make the dataset more reliable, non-textual symbols and characters were removed so as to implement language models for the Roman Urdu analysis and evaluation of proposed model. The sample sentences from the dataset are shown in Table I with English translation for better understanding the script style of Roman Urdu.

Lexical Variation in Preprocessed Dataset: The Roman script does not follow any standard which makes it more complicated than English language dataset. Different spelling

*https://archive.ics.uci.edu/ml/datasets/Roman+Urdu+Data+Set

refer to same word and identical spelling refer to different contextual words. This phenomenon confuses the embedding of vocabulary and motivated us for normalization of Roman Urdu sentences. Previously, some approaches have been used for normalization purposes to reduce the variation of embedding for the same word: Urdu phone, Similarity function, Lex-C clustering algorithm, Stemming and Lemmatizing [27], [28]. These approaches depend upon some rules and there is 30% to 40% chance of failure attributed to these rules. Making a set of similar words and clipping suffix or prefix can negatively influence the embedding behavior towards sentence polarity.

Standards were followed to apply lexical normalization and standardization of words. As discussed in [29] each word of the Urdu language should follow the standard spelling as it comes to Roman transliteration of Urdu terms. Unification of vocabulary (where each word refers by unique characters not multiple combinations of characters) was done by same person to maintain the consistency for the whole dataset.

*2) **Normalized Dataset**:* From the preprocessed dataset, the sentences that have more polarized words are normalized manually for unification of vocabulary. Three different categories were mainly focused of preprocessed to normalize. For the sake of generalization and avoiding overfitting, sentences belong to different categories from different sources are normalized. These categories include news, reviews about celebrities and feedback about products received through online shopping. All of these categories have equal number of positive and negative sentences.

TABLE II. LEXICAL VARIATION OF WORDS (FROM ROMAN URDU SENTENCES IN TABLEI)

English	Preprocessed Roman Urdu	Normalized Roman Urdu
Successful	kamiyaab, kamyab, kaamyaab, kamyaab	kamyaab
Beautiful	khubsurat, khubsoorat, khoobsurat, khoobsoorat	khoobsoorat
Good	acha, achi, ache	same as in preprocessed
Bad	bura, buri, bure	same as in preprocessed

The lexical variation of the words is represented in Table II that influences the polarity of the sentence. The Roman Urdu terms *kamyaab, kamiyaab, kamyab, kaamyaab* for *successful* and *khoobsoorat, khubsurat, khubsoorat, khoobsurat* for *beautiful* in first two sentences of Table I are normalized to *kamyaab* and *khoobsoorat* respectively as shown in Table II. Roman Urdu terms *acha, achi, ache* for *Good* and *bura, buri, bure* for *Bad* in the next two sentences of Table I depend upon the gender and number of subject word (singular or plural). Therefore, these terms are not normalized and will remain the same as in preprocessed dataset. Even though the normalization process increases the accuracy as mentioned in [30] but the existence of this limitation in the Urdu language leads to produce low results as compared to other languages. The resultant dataset called as normalized dataset. Considering the time limit and assessing the performance improvement of model, we normalized 3000 Sentences.

C. Effect of Normalization

The similarity of embedding-vectors measured by cosine distance sort the words in the vocabulary according to their

TABLE III. SIMILARITY %AGE IN NORMALIZED DATASET DENOTED BY NORM %SIM IN THE TABLE AND SIMILARITY %AGE IN PREPROCESSED DENOTED BY PREPROC %SIM IN TABLE

(A) THIS TABLE REPRESENTS THE SIMILARITY PERCENTAGE OF SIMILAR WORDS THAT BELONG TO SAME CLASS.

Similar word	Norm %Sim	Preproc %Sim
Pyar;sakoon; [love; calm]	**99**	92
Qeeemati;khoobsoorat; [Expensive; Beautiful]	**98**	95
Shohrat;Fatah; [Fame; Victory]	**99**	96

(B) THIS TABLE REPRESENTS THE SIMILARITY PERCENTAGE OF DISSIMILAR WORDS THAT BELONG TO DIFFERENT CLASS.

Dissimilar words	Norm %Sim	Preproc %Sim
Zakhmi;sehat; [Injured; Health]	**70**	75
Janbahaq;zinda; [Died; live]	**58**	62
Shohrat;badnam; [Fame; disgrace]	**71**	76

"similarity" in the embedding-space. Table III (a) indicates that similar embedding vectors of different words belonging to the same class (have same contextual meanings) appeared to give high results. Table III (b) indicates that words have different or opposite contextual meaning belonging to different classes must indicate less similarity and give low results. Tables are prepared of the same word for preprocessed (unnormalized) and normalized dataset. From results, it can be observed that similar words belonging to the vocabulary of normalized dataset show higher similarity than a preprocessed dataset. Those words that are contextually less similar show lower results for the vocabulary of a normalized dataset. Reason lies in the unification and Normalization of terms. A different variation of a word creates ambiguity that leads the network to become less efficient despite having a large number of sentences to train the model. On the other case, a small dataset with unique terms and no multiple variations for the same word make the dataset consistent on which network produces better results. For example, word successful *kamyaab* has multiple variations *kamiyaab*, *kamyab*, *kaamyaab* in Roman Urdu that has changed to one term *kamyaab* in the normalized dataset. This unique word has the highest similarity with itself as compare to different variation, this reason influences the results considerably.

V. RESULTS AND ANALYSIS

This section exhibits the results of all language models evaluated on preprocessed and normalized Roman Urdu sentences. Comparison of experimental results, finding and contributions are discussed. Table IV and Table V contain multiple matrices including testing accuracy, recall (true positive rate) and precision (positive predicted value) to assess the efficiency of each language model as shown in equations (10), (11), and (12) respectively. Moreover, time complexity is represented as subscript of testing accuracy to show the time taken by each experiment for corresponding language model. Besides, accuracy on testing dataset and utilization of time resources, recall and precision show the exactness(quality) and completeness(quantity) of each language model.

$$Accuracy = \frac{tp + tn}{(tp + tn + fp + fn)} \tag{10}$$

$$Recall = \frac{tp}{(tp + fn)} \tag{11}$$

$$Precision = \frac{tp}{(tp + fp)} \tag{12}$$

The results demonstrate two aspects for the evaluation of the language models. First, the performance measure matrices including recall, precision, and accuracy. Second, the time in seconds represents the time complexity of the experiment. The slight difference in results for different evaluation metric indicates the consistent performance of the model.

TABLE IV. EXPERIMENTAL RESULTS ON PREPROCESSED DATASET.

Language models	Recall	Precision	$Accuracy_{seconds}$
Fasttext	63.8	62.3	$62.4_{(220)}$
CNN	60.6	60.6	$60.6_{(400)}$
LSTM	66.2	66.4	$66.2_{(362)}$
BiLSTM	66.9	67.0	$66.9_{(465)}$
Self-Attention	66.9	66.6	$66.8_{(230)}$
SA-BiLSTM	68.5	68.4	$\mathbf{68.4}_{(260)}$

Results in the Table IV confirm these findings: CNN is least efficient in accuracy and time complexity as CNN does not extracts the important embedding from sentences as compared to Attention mechanism. Fasttext is an agile network and produce better results than CNN but still it is far low than other models as Fasttext is not as deep. The results produced by LSTM and BiLSTM (RNN variants) on Roman Urdu sentences is remarkably high than former networks. However, limitation of these methods are that they consume high time and memory cost. Results depicts that Self-attention is efficient in time memory complexity and achieves comparable accuracy with LSTM, BiLSTM. Therefore, it is generally accepted that Self-Attention addresses the issues arise in other RNN variants.

The proposed network, SA-BiLSTM outperformed all neural network by achieving highest accuracy of 68.4%. From the results, it is clear that proposed network utilized less time resources than CNN, LSTM and BiLSTM. These results support the effectiveness of model by attaining better outcome on all matrices when compared with other language models. The selective and bidirectional architecture of SA-BiLSTM results in the highest accuracy for complex sentence structure of Roman Urdu script possessing lexical variation of words.

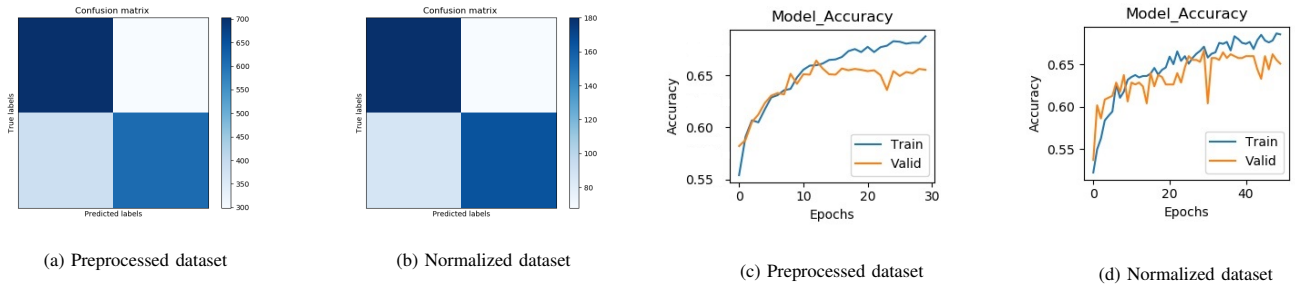
TABLE V. EXPERIMENTAL RESULTS ON NORMALIZED DATASET.

Language models	Recall	Precision	$Accuracy_{seconds}$
Fasttext	62.1	62.2	$62.1_{(100)}$
CNN	64.6	65.0	$64.6_{(214)}$
LSTM	67.8	68.1	$67.8_{(150)}$
BiLSTM	67.7	67.9	$67.6_{(240)}$
Self-Attention	67.2	67.1	$67.0_{(90)}$
SA-BiLSTM	69.4	69.3	$\mathbf{69.3}_{(120)}$

Normalized dataset is 73% smaller in size than preprocessed dataset in terms of sentences and pre-trained word vectors. In spite of this fact, all language models have produced better results on this dataset even if the improvement is negligible as shown in Table V. Contrary to Previous experiments, CNN has yielded higher accuracy than Fasttext which shows that CNN performs well on Normalized dataset. The results produced by LSTM, BiLSTM and Self-Attention on Normalized dataset are in line with trend of results on preprocessed dataset. The proposed model delivered significantly better results for all matrices and highest accuracy i.e. 69%. It can be seen that SA-BiLSTM supersedes the existing models in all metrics. Even though deep learning models achieved adequate results, the limitation we faced thoroughly in the experiments was the unavailability of large pre-trained word embeddings due to the absence of a massive dataset like Google News or Wikipedia. As the previous study mentioned that less pre-train embedding did not produce good results, despite, the performance of SA-BiLSTM on Roman Urdu dataset is in line state-of-the-artwork.

Additionally, these results endorse our claim that consistent dataset with more polarized sentences, having normalized vocabulary can produce more efficient results, although it has trained on a smaller number of sentences and pre-trained word embedding. The confusion matrix shown in Fig. 3 upholds the normal behavior of proposed model. From Fig. 3a and 3b, it is obvious that SA-BiLSTM succeeded in detecting true positive and true negative by giving strong confusion matrix for normalized dataset.
Fig. 3 expresses the accuracy curves of model on preprocessed and normalized dataset respectively. The accuracy curve on training and validation set of data represent the learning and generalizing ability of SA-BiLSTM. For the case in Fig. 3c, the accuracy curve on validation data displays that the experiment stopped earlier (in 30 epochs). The curve hits maximum accuracy of 67% for training and and 65% for validation. The existence of noise in embedding space (in the preprocessed dataset), restricts the model to learn after reaching at certain limit (65%), even with more number of sentences and a large number of pre-trained embedding. It can be seen in Fig. 3d that the model accuracy curves enter in stabilized region after 40 epochs. It depicts that the model learns the features smoothly in more than 40 epochs. Even though the less number of sentences and short vocabulary to create pre-trained embeddings, training and validation curves of accuracy reached 70% and 68% respectively. Moreover, despite the complexities in processing Roman Urdu (as explained earlier) the difference between the training and validation curves in Fig. 3d is less than 2%, which is well in line for a models to be considered as a good fitted model. This upholds the validation of model on normalized dataset.

Fig. 4 illustrate the result summary of language models on both datasets. Accuracy on the normalized dataset is higher for each language model as compare to the preprocessed dataset. From figure 4, it must be pointed out that results are getting increasingly better from CNN to SA-BiLSTM on the normalized dataset. Experimental results prove that every model performs better on normalized dataset (which is 3x times smaller) than preprocessed dataset and utilizes less time cost. SA-BiLSTM results in the highest



(a) Preprocessed dataset (b) Normalized dataset (c) Preprocessed dataset (d) Normalized dataset

Fig. 3. Confusion matrix and Accuracy curves of SA-BiLSTM

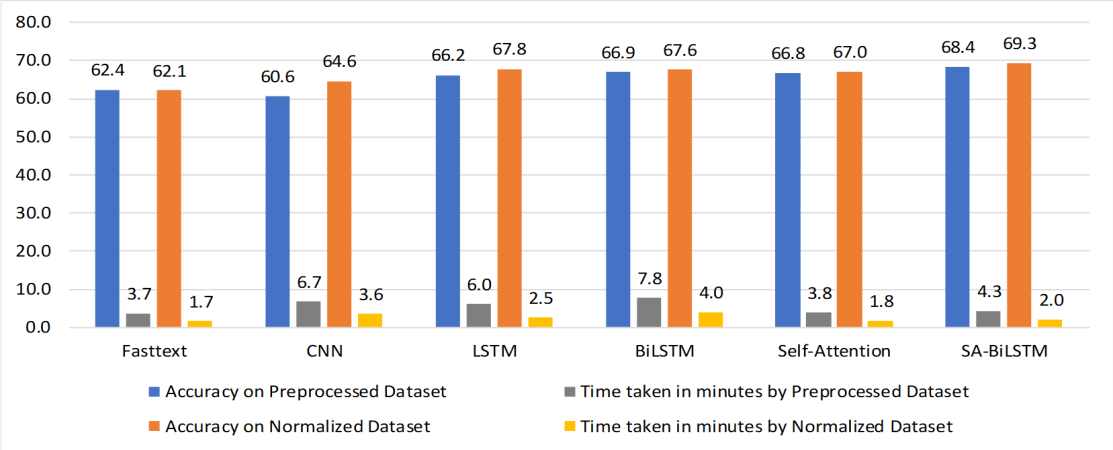


Fig. 4. Comparison of Accuracy achieved by all language models w.r.t time (minutes) on both dataset

accuracy because of its selective and bidirectional nature on both datasets which confirm that this integrated model is the best choice for sentiment analysis of Roman Urdu. Additionally, Normalization of the dataset is important for all non-English languages as it improves the performance of the model.

VI. Conclusion

In this paper, we present a novel deep learning model for sentiment analysis of Roman Urdu. This particular script hinders direct approaches owing to its complex sentence structure, and numerous lexical meaning. Proposed model utilizes the traits of Self-attention and Bidirectional LSTM (SA-BiLSTM) network to yields better results. Moreover, to make a fair comparison, we preprocessed and normalized the dataset. Experimental results indicate that SA-BiLSTM surpasses existing deep learning models in accuracy and requires fewer resources. SA-BiLSTM achieves a high accuracy of 68.4% and 69.3% for preprocessed and normalized datasets, respectively.
As for future research, we can try to enhance the efficiency of SA-BiLSTM and bring it to use for language inference and generation tasks, and these are very critical components to increase normalized vocabulary, vast pre-trained embedding, and massive datasets for better analysis.

References

[1] B. Liu, *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press, 2015.

[2] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[3] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[4] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *arXiv preprint arXiv:1503.00075*, 2015.

[5] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[7] N. Parmar, A. Vaswani, J. Uszkoreit, Ł. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," *arXiv preprint arXiv:1802.05751*, 2018.

[8] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, "Generating wikipedia by summarizing long sequences," *arXiv preprint arXiv:1801.10198*, 2018.

[9] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "Disan: Directional self-attention network for rnn/cnn-free language understanding," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[10] M. V. Mäntylä, D. Graziotin, and M. Kuutila, "The evolution of sentiment analysis—a review of research topics, venues, and top cited papers," *Computer Science Review*, vol. 27, pp. 16–32, 2018.

[11] K. Ravi and V. Ravi, "Sentiment classification of hinglish text," in *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*. IEEE, 2016, pp. 641–645.

[12] H. Kaur, V. Mangat, and N. Krail, "Dictionary based sentiment analysis of hinglish text," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, 2017.

[13] X. Ouyang, P. Zhou, C. H. Li, and L. Liu, "Sentiment analysis using convolutional neural network," in *2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing*. IEEE, 2015, pp. 2359–2364.

[14] C. Zhou, C. Sun, Z. Liu, and F. Lau, "A c-lstm neural network for text classification," *arXiv preprint arXiv:1511.08630*, 2015.

[15] P. Wang, J. Xu, B. Xu, C. Liu, H. Zhang, F. Wang, and H. Hao, "Semantic clustering and convolutional neural network for short text categorization," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2015, pp. 352–357.

[16] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III, "Deep unordered composition rivals syntactic methods for text classification," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2015, pp. 1681–1691.

[17] Y. Goldberg, "A primer on neural network models for natural language processing," *Journal of Artificial Intelligence Research*, vol. 57, pp. 345–420, 2016.

[18] J. Islam and Y. Zhang, "Visual sentiment analysis for social images using transfer learning approach," in *2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom)*. IEEE, 2016, pp. 124–130.

[19] L. Yanmei and C. Yuda, "Research on chinese micro-blog sentiment analysis based on deep learning," in *2015 8th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 1. IEEE, 2015, pp. 358–361.

[20] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[21] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[24] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using bilstm-crf and cnn," *Expert Systems with Applications*, vol. 72, pp. 221–230, 2017.

[25] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3429–3437.

[26] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[27] Z. Sharf and S. U. Rahman, "Lexical normalization of roman urdu text," *International Journal of Computer Science and Network Security*, vol. 17, no. 12, pp. 213–221, 2017.

[28] A. Rafae, A. Qayyum, M. Moeenuddin, A. Karim, H. Sajjad, and F. Kamiran, "An unsupervised method for discovering lexical variations in roman urdu informal text," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 823–828.

[29] T. Ahmed, "Roman to urdu transliteration using wordlist," in *Proceedings of the Conference on Language and Technology*, vol. 305, 2009, p. 309.

[30] R. Satapathy, C. Guerreiro, I. Chaturvedi, and E. Cambria, "Phonetic-based microtext normalization for twitter sentiment analysis," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2017, pp. 407–413.