ABSTRACT

Although over 169 million people in the world are familiar with the Urdu language and a large quantity of Urdu data is being generated on different social websites daily, very few research studies and efforts have been completed to build language resources for the Urdu language and examine user sentiments. The primary objective of this study is twofold: (1) develop a benchmark dataset for resource-deprived Urdu language for sentiment analysis and (2) evaluate various machine and deep learning algorithms for sentiment. To find the best technique, we compare two modes of text representation: count-based, where the text is represented using word \( n \)-gram feature vectors and the second one is based on fastText pre-trained word embeddings for Urdu. We consider a set of machine learning classifiers (RF, NB, SVM, AdaBoost, MLP, LR) and deep leaning classifiers (1D-CNN and LSTM) to run the experiments for all the feature types. Our study shows that the combination of word \( n \)-gram features with LR outperformed other classifiers for sentiment analysis task, obtaining the highest \( F_1 \) score of 82.05% using combination of features.

INDEX TERMS

Urdu sentiment analysis, machine learning, deep learning, natural language processing.

I. INTRODUCTION

In recent years, with the remarkable increase in the use of hand-held devices and the internet, the use of social media such as Twitter, Facebook, and blogs has been equally increasing by individual users to express their emotions and sentiments [1]–[3]. Currently, people want to publicly share their opinions, feedback, reviews, and feelings about products, politics, or any viral news. As a results, businesses and institutes are searching for useful information from social media [4]–[7]. Therefore, there is a need for intelligent systems such as sentiment analyzers, which can convert raw social media user data into useful information. To recognize and detect emotions and for sentiment analysis, languages such as English, French, German, and other European languages are considered rich languages in terms of tool accessibility. Nevertheless, languages such as Urdu, Punjabi, and Hindi are judged resource deprived [8]. Urdu is very different from other languages due to its morphological structure as it starts from right to left. Due to its morphological structure, the Urdu script is not very common; therefore, a standard dataset or corpora is required to perform natural language processing tasks.

Sentiment analysis of the Urdu language is equally essential, as it is important in other languages, as it assists non Urdu speakers in grasping the basic feelings, emotions, and opinions of any user behind text. Urdu is the national and official language of Pakistan and commonly spoken medium in many states of India.\(^1\) In regard to social media, several native Urdu speakers use Urdu script on platforms such as Twitter, Facebook, and YouTube to express their emotions, feelings, and opinions. As a result, it is important to analyze the Urdu text to understand the opinions and feelings of native Urdu speakers.

There are many problems with Urdu sentiment analysis, such as a shortage of recognized lexical resources [9]–[11].
Mostly, Urdu websites are developed in a descriptive arrangement rather than a proper text encoding structure; due to this hurdle, it is challenging to create a benchmark corpus in Urdu. Urdu sentiment analysis has not yet been investigated completely even after its considerable use; most of the existing literature studies are focused on different aspects of language processing [12], [13].

In this paper, the primary focus is to contribute a benchmark corpus for Urdu sentiment analysis. Our corpus known as Urdu Corpus for Sentiment Analysis (UCSA). This new dataset and experiments provide a benchmark enabling further research in sentiment analysis in Urdu language.

The main contributions of this research are as follows:

- A new sentiment analysis corpus in Urdu is collected that contains user reviews about various services: products, games, and politics. It is manually annotated by experts following a set of guidelines (publicly available; see a link below);
- We provided baseline results for the state-of-the-art machine leaning (RF, NB, SVM, AdaBoost, MLP, LR) and deep learning (1D-CNN, LSTM) models on our UCSA corpus using two text representations: word n-gram features and fastText pre-trained word embeddings;
- To the best of our knowledge, no research study shows the use of deep learning models with pre-trained word embedding models for Urdu sentiment analysis; therefore, we studied the effectiveness of word embedding models in resource-deprived languages such as Urdu.

Our corpus UCSA is publicly available.²

The rest of the paper is organized as follows. Section II presents the background and related work. Section III describes the corpus collection details. Section IV presents the methodology of the paper. Section V analyzes the experimental setting and results. Finally, Section VI concludes the paper.

II. BACKGROUND AND RELATED WORK

In this section, we discuss famous datasets as well as machine and deep learning techniques for sentiment analysis.

A. SENTIMENT ANALYSIS DATASETS AND TECHNIQUES

To create a benchmark dataset for sentiment analysis, SemEval contests are considered one of the most noticeable literature efforts. In the series of SemEval competitions to examine sentiment analysis, researchers performed distinct tasks using different datasets. These datasets developed in Arabic and English [14]. Generally, these datasets contain user tweets from Twitter and they are related to different products such as laptops, TVs, and mobiles. The SemEval corpus 2013 edition consists of Twitter and SMS data; Twitter tweets were divided into three sets: training (9,728), development (1,654) and test (3,813) while SMS messages were used for testing purpose only, which contains 2,093 messages.

²http://ieeexplore.ieee.org/document/7804; last visited: 20-06-2021

Similarly, the 2014 version of the SemEval Twitter dataset contains 1,853 user tweets and 1,142 LiveJournal news [15]. The 2016 and 2017 versions of the SemEval datasets were split into training, development, and test sets for each subtask [16]. In this edition there were five subtasks: A, B, C, D, and E.

In addition to the SemEval efforts, Korean, German, and Indonesian languages have also been investigated for sentiment analysis. A Korean dataset was created (KOSAC) that contains 332 news articles. Their primary aim was to examine sentiments in Korean and they used Korean subjectivity markup language to annotate their dataset. Another dataset was developed that contains customer reviews about various Amazon products [18]. Amazon review parser was used for the dataset collection. Human experts annotated each review according to their semantic meaning. A total of 63,067 reviews were collected about different products. Another effort was made to develop an Indonesian corpus. The Twitter Streaming API was used to collect this dataset and they also used geo location just to collect Indonesian dialect tweets. Their Indonesian dataset contains 5.3 million tweets [19].

Recently, deep learning methods were implemented to investigate text representations and to overcome the problem of sentiment classification on a large social network datasets [20]–[22]. In addition, improved word vectors (IWWVs), was recommended for word embedding because of their higher performance in the domain of sentiment analysis [23].

A few study were performed on the sentiment analysis of social network data on the subject to support intelligent transportation systems [24]–[26]. Data were gathered from various social networking sites such as Facebook, Twitter, TripAdvisor. They achieved an accuracy of 93% on their sentiment analysis dataset. In addition, based on social network data, a real-time observation framework was suggested to detect traffic accidents and analyze traffic conditions by using BiLSTM [26]. They achieved an accuracy of 97% for traffic event detection analysis.

B. URDU DATASETS FOR SENTIMENT ANALYSIS

Although a considerable quantity of data is available on internet research on sentiment analysis, Urdu is still at the initial level compared to other resource-rich languages such as English. A large quantity of data is required to create a benchmark dataset for sentiment analysis. The drawbacks of existing corpora are that they are too small or contains data about limited genres.

In the first study [27], authors collected user reviews to create two corpora to find their models efficiency. The first corpus contains 322 positive and 328 negative movie reviews. The second corpus contains reviews about electronic appliances. This dataset contains 650 user reviews, among 322 are positive and 328 are negative. In this study, they used grammatical-based approach as well as they focused on sentence grammatical structure. They achieved 82.5%
accuracy on their best model. There are many problems with their dataset as they did not mention any data annotation techniques as well as their corpus is not publicly available.

In another study [28], authors extracted Urdu text from Urdu news websites such as BBC Urdu news and Dawn news on a particular topic for corpus generation. The authors used a lexicon-based architecture and assigned polarity to each token according to its sentiment. To find the model efficiency, they performed experiments only on 124 comments, which they extracted from different websites. The lexicon-based model reveals an overall accuracy of 66%. The most significant effort to build an Urdu sentiment analysis corpus was made by the authors of study [27]. This study began with a collection of Urdu blogs of different genres. A total of 6,025 Urdu sentences were gathered from 151 different blogs. Three human experts annotated these collected sentences into positive, negative, and neutral classes. After applying basic pre-processing techniques such as stop word removal, the authors used LIBSVM library to implement decision tree (DT), and k-nearest neighbors algorithm (k-NN) algorithms for classification purpose. They achieved highest accuracy of 67.01% on k-NN classifier. The corpus used in this study is not publicly available.

Note that Urdu is a resource-deprived language, linguistically and technically. According to the existing literature, many of the procedures applicable to sentiment analysis of other languages are not relevant to the Urdu language due to morphological structure [29], [30]. Additionally, the deficiency in linguistic and language resources such as lexicons and corpora also makes it difficult to implement the currently available sentiment analysis methods cited in the literature review, such as the availability of lexicons and datasets. Moreover, accessible annotated datasets are not sufficient for implementing useful sentiment analysis. In addition, datasets and sentences generally belong to the same or limited genres. To reduce this deficiency, this study emphasizes building an Urdu dataset containing sentences fitted to six different domains. We implemented machine learning and deep learning models on our constructed corpus, UCSA, which has not yet been studied fully for the sentiment analysis of Urdu data.

### III. BUILDING THE DATASET

This section describes the procedures to create an annotated Urdu dataset for sentiment analysis. The stages included for building the Urdu corpus are collecting user reviews from the internet, preparation of annotation rules, manual annotation, and final version of the corpus.

#### A. COLLECTING REVIEWS FROM THE INTERNET

To build a benchmark dataset for Urdu sentiment analysis, user reviews contain information about various services, products, games, and politics from different websites that allow users to post their Urdu views. Urdu is a resource-deprived language; therefore, the authors decided to collect data about different genres from internet repositories that are easily accessible to construct a standard Urdu language text corpus. Consumer reviews contain information about politics, movies, Urdu drama, TV talk shows and sports. Four individuals were hired for manual data collection. They were native Urdu speakers and it took 3 months to collect the raw data. Initially, data was gathered in an Excel sheet.

#### B. ANNOTATIONS GUIDELINES

This section explains the annotation process that the authors used in manual corpus generation. This step includes preparing the rules or guidelines for annotation, manual annotation of the complete dataset by native Urdu speakers. We design rules for sentiment analysis from existing literature review. Figure 1 shows examples of user reviews fitting to the positive and negative classes.

- A sentence is labeled as positive if it conveys an overall positive sentiment or if it expresses both positive and neutral or if it contains agreement approval [31], [32];
- Sentences with words such as congratulations and admiration were also marked as positive [32];
- A sentence is labeled as negative if it conveys an overall negative sentiment or if it has more negative words than the other sentiments [19];
- If any sentence shows any disagreement, then the sentence is classified as negative [32];
- If a sentence has terms such as ban, penalizing, and assessing, it is labeled negative [32];
- If a sentence comprises a negative word with a positive adjective, it is classified as negative [33].

#### C. DATASET STATISTICS

The Urdu dataset was manually annotated by three (X, Y, and Z) human experts to create a benchmark dataset hereafter named Urdu Corpus for Sentiment Analysis (UCSA). Native Urdu speakers annotated all the user reviews, and all were master graduates in the Urdu language. The annotators were aware of sentiment analysis and annotation rules, as discussed above. Experts X and Y annotated each sentence either in positive or negative classes, considering the above-discussed rules. The conflicts between X and Y were resolved by Z by labeling the review. We obtained an
inter-annotator agreement (IAA) of 73.91% and a Cohen Kappa score of 59.7% (moderate) on our UCSA dataset. IAA and moderate scores revealed that the annotators followed the annotation guidelines during the labeling phase. UCSA contains 9,601 user reviews, of which 4,843 are positive and the remaining are negative reviews, as shown in Table 1. From the statistics in Table 1, it can be clearly seen that our corpus is class balanced. Very few scholars in the existing literature have made efforts to create datasets for carrying out experiments. Nevertheless, unluckily, most of the currently available datasets are very small and are from specific genres or cover very few genres rather than different genres. The corpora [19], [27] are small and contain user reviews to specific fields.

### IV. METHODOLOGY

This section focuses on the experimental details of our machine learning and deep learning models such as the support vector machine (SVM), naïve Bayes (NB), random forest (RF), AdaBoost, multilayer perceptron (MLP), logistic regression (LR), 1-dimensional convolutional neural network (1D-CNN), and long short-term memory (LSTM). All these machine and deep learning models have been implemented on our proposed UCSA corpus. Figure 2 represents the overall architecture of the system.

#### A. PREPROCESSING

The preprocessing of Urdu text is essential to make it easy and useful for NLP tasks. To enhance our model’s accuracy, emoji’s, URLs, email addresses, phone numbers, numerical numbers, numerical digits, currency symbols, and punctuation marks were removed. Additionally, the following text preprocessing steps were performed to increase our model’s effectiveness for Urdu text.

1) STOP WORDS

The words used to complete sentences are called stop words. Words such as “لا” and “ما” are commonly used words in Urdu. We removed these words from our corpus. Nevertheless, due to the Urdu language’s morphological structure and poor resources, it is challenging to remove stop words automatically. Figure 3 explains the flowchart of the Urdu stop words removal steps. All commonly used Urdu stop words were collected in a file, and then all those were eliminated from the corpus.

2) NORMALIZATION

The normalization of Urdu text is essential to make it advantageous for NLP-related tasks. This step is used to solve the issue of correct encoding for the Urdu characters. Normalization is used to obtain all the characters in the required unicode range (0600-06FF) for Urdu text. This step is also used to avoid the concatenation of different Urdu words. For example, “خوش بانی” is one word (unigram) with two different strings. These two strings (khus and bash) are part of the same word concerning syntax and semantics. If the space between two strings is omitted, then we obtain “خوشبانی” which is an incorrect word in the Urdu language. With the help of normalization, authors attempt to minimize this effect. We used UrduHack library for this task.

#### B. N-GRAM FEATURES

In natural processing tasks such as text classification, the text is generally denoted as a vector of weighted features. In this study, different n-gram models are used; these are the models
that allocate probabilities to a sequence of words. An \( n \)-gram is a sequence of \( n \) words; a unigram is a model that contains a sequence of one word such as “homework”; similarly, a bigram is a sequence of two words such as “your homework” and a trigram model contains a sequence of three words such as “complete your homework”. We explored \( n \)-gram features such as unigram, bigram and trigram on our dataset.

C. PRE-TRAINED WORD EMBEDDINGS

Recently, pre-trained word vector models have been applied in many natural processing tasks and have shown state-of-the-art results. The basic concept behind these pre-trained models is to train these models on very large corpora and fine tune these models for specific tasks. fastText [33] is a word vector model trained on Wikipedia and common crawl datasets. This model is trained for a total of 157 languages, including Urdu. This is the motive behind using the fastText word embedding model for this task with deep learning models. The fastText model was trained using skip-gram and continuous bag of words (CBOW) [34], [35]. The fastText model is an extension of skip-gram that breaks down the unigram (words) into bags of character \( n \)-grams (sub-words) and allocates a vector value to individual character \( n \)-grams. Therefore, each single word is represented by the summation of its related \( n \)-gram vectors.

D. CLASSIFICATION MODELS

Various machine and deep learning models, namely, SVM, NB, RF, AdaBoost, MLP, LR, 1D-CNN, and LSTM are used to find the effectiveness of our corpus and achieve state-of-the-art results. We do not explain these conventional machine learning models here because these models are prevalent and famous. Two deep-learning classifiers were applied to find the best performing classifier on our dataset such as 1D-CNN and LSTM. Keras neural-network library\(^4\) was used for the implementation of 1D-CNN and LSTM to implement the state-of-the-art baseline approaches for the comprehensive evaluation of sentiment analysis on our dataset.

Primarily, 1D-CNN is used for the computer vision, however, it performs well on classification tasks in natural language processing domain. A 1D-CNN is extremely capable when you expect to acquire new attributes from short fixed-length chunks of the overall data set and where the placement of the feature is not of relevance [36]–[38]. LSTM [39] is recurrent neural network architecture and shows state-of-the-art results for sequential data. Basically, LSTM is designed to capture the long-term dependencies between text data. For each time step, the LSTM model obtains the input from the current word, and the output from the previous or last word produces an output, which is used to feed to the next state. The hidden layer from the previous state (and sometimes all hidden layers) is then used for classification. The high-level system architecture of an LSTM network with fastText embedding is shown in Figure 4. A typical LSTM network contains four main components: input gate, forget gate, memory cell, and output gate. Basically, these gates are used to flow in and out of the data at the existing time step. LSTM working is divided into three parts as follow:

1) STEP 1

LSTM identify the insignificant information in the first step and disappear it from the cell. Sigmoid layer is used for the identification and elimination of details by acquiring output from the final LSTM unit \( h_t - 1 \) at time \( t-1 \) and the available input \( X_t \) at time \( t \), sigmoid function clarify which chunk of old output should be removed. The output should be in the range between 0 and 1, which is stored in the vector \( f_t \), for every cell state \( C_t - 1 \). Sigmoid function takes the decision that information should be kept or discarded based on the output.

\[
f_t = \sigma(W_f[h_t - 1, X_t] + b_f)
\] (1)

---

\(^4\)https://keras.io/
2) STEP 2
In step 2, we store new input $X_t$ as well as update the cell state. we executes two actions: one is for sigmoid layer while other one is for tanh layer. Sigmoid layer makes a decision which information need to be update or discard while tanh layer allocate weights to the passing values. Then these values are multiplied to update the cell state and then add new memory to old memory $Y_{t-1}$ that result in $Y_t$ [2].

$$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i),$$  \hspace{1cm} (2)

$$N_t = \tanh(W_n[h_{t-1}, X_t] + b_n),$$  \hspace{1cm} (3)

$$Y_t = Y_{t-1} \ast f_t + N_t \ast i_t.$$  \hspace{1cm} (4)
TABLE 4. Comparison with existing state-of-the-art results.

| Classifiers            | Acc  | P    | R    | F₁   |
|------------------------|------|------|------|------|
| k-NN [20]              | 67.01| 67.40| 67.03| 67.03|
| LIBSVM [20]            | 65.00| 67.43| 65.00| 64.57|
| Lexicon-based SVM [19] | 66.00| 69.00| 79.00| 73.00|
| SVM (Our experiments)  | 81.47| 80.32| 82.36| 81.47|
| RF (Our experiments)   | 75.27| 71.36| 81.30| 74.85|
| NB (Our experiments)   | 78.99| 76.80| 81.83| 78.99|
| AdaBoost (Our experiments) | 79.82| 77.98| 81.94| 79.91|
| MLP (Our experiments)  | 80.44| 80.36| 79.51| 79.93|
| LR (Our experiments)   | 81.94| 79.95| 84.26| 82.05|
| 1D-CNN (Our experiments) | 75.73| 78.43| 76.78| 75.57|
| LSTM (Our experiments) | 75.96| 77.76| 77.83| 77.52|

where $Y_t – 1$ and $Y_t$ are showing the cell states at time $t – 1$ and $t$. While $W$ represent weight matrices and $b$ represent bias to the cell state.

3) STEP 3
In this step, we have output values $h_t$. These values based on output cell state $Y_t$; however, in a filtered form. The last step is related to output values $h_t$, which depend on the output cell state $Y_t$ but in a filtered form. In ordered to create output, sigmoid layer choose the part of cell state. After that sigmoid gate $Y_t$ output is multiplied by the new values that are produced by tanh layer from the cell state $Y_t$

$$Y_t = \sigma (W_o [h_t – 1, X_t] + b_o)$$  \hspace{1cm} (5)
$$h_t = Y_t \ast \tanh(C_t)$$  \hspace{1cm} (6)

$W_o$ and $b_o$ depicts the weight matrices and bias.
E. EVALUATION MEASURES
We evaluate the effectiveness of our sentiment analysis models using Recall (R), Precision (P), and F₁-measure. The mathematical equations are as follows:

\[
\text{Precision} = \frac{TP}{TP + FP},
\]

\[
\text{Recall} = \frac{TP}{TP + FN},
\]

\[
F_1 = \frac{2 \times P \times R}{P + R},
\]

where TP and FP stand for true positive and false positive, and FN stands for false negative.

V. EXPERIMENTAL SETTINGS AND RESULTS
We performed our experiments on UCSA, which is publicly available to the research community. UCSA contains 9,601 Urdu reviews, which belong to politics, dramas, movies, TV talk shows, sports, and software domains. The dataset is split into training, which contains 80% of user reviews, and testing, which contains 20%. In all the experiments for machine learning models we used default
parameters. For deep learning algorithms, we used mean square error (MSE) as a loss function, Adam as an optimizer. We set the number of epochs to 25.

**A. RESULT AND DISCUSSION**

Each of the six machine learning classifiers is run on the UCSA dataset using word n-gram features. All the revealed results are carefully examined to improve the results and identify the finest machine learning classifier with features that achieve better results than the others concerning the accuracy, precision, recall, and $F_1$ score. By witnessing the Table 2 results, all the machine learning classifiers’ performances are quite poor with the trigram feature. Generally, there are discriminative models (SVM, LR, etc.) and generative classification models (NB).

Both SVM and LR achieve satisfactory results, as both classifiers belong to discriminative models. Logistic regression is a supervised machine learning algorithm that is used when problems are categorical in nature and it is the most commonly used classifier when the data have two classes, either positive or negative. Overall, the highest accuracy of 81.94%, precision of 79.95%, recall of 84.26%, and $F_1$ score of 82.05% were achieved by LR with the combination of n-gram features. The SVM classifier achieves the second highest accuracy, precision, recall and $F_1$ score, which were 81.47%, 80.32%, 82.36%, and 81.47%, respectively, with the unigram feature.

The worst accuracy out of all classifiers was 55.25% gain by RF with trigram features. All classifiers perform better with bigram features as compared to trigram features. The overall results using different machine learning models with different features shown in Table 2. Figures 5, 6, 7, 8 and 9 describe the comparison of each model in terms of accuracy, precision, recall and $F_1$ measure with word n-gram features.

Table 3 presents the results of deep learning models on our dataset. LSTM achieves slightly better results than the 1D-CNN model in terms of accuracy, which is 75.96 for LSTM and 75.73% for 1D-CNN. Deep learning results are slightly lower than machine learning models. It is because some of the words are out of vocabulary in fastText pre-trained model. Therefore, in machine and deep learning our results are in line with state-of-the-art results.

As previously stated, a lack of research using machine learning algorithms in Urdu sentiment analysis is seen. Very few studies are found regarding this context and they used different machine learning classifiers on a very insignificant dataset. Our dataset contains more user reviews as a comparison to previous studies. The results of our study reveal that each model in our study performs better than existing models. A comparison of our study with existing studies is presented in Table 4.

**REFERENCES**

[1] S.-U. Hassan, A. Akram, and P. Haddawy, “Identifying important citations using contextual information from full text,” in *Proc. ACM/IIEEE Joint Conf. Digit. Libraries* (JCDL), Jun. 2017, pp. 1–8.

[2] Y. Liu, F. Du, J. Sun, T. Silva, Y. Jiang, and T. Zhu, “Identifying social roles using heterogeneous features in online social networks,” *J. Assoc. Inf. Sci. Technol.*, vol. 70, pp. 660–674, Mar. 2019.

[3] Z. Luo, S. Huang, and K. Q. Zhu, “Knowledge empowered prominent aspect extraction from product reviews,” *Inf. Process. Manage.*, vol. 56, no. 3, pp. 408–423, May 2019.

[4] F. Anwar, N. Ittai, H. Afzal, and R. Nawaz, “HRS-CE: A hybrid framework to integrate content embeddings in recommender systems for cold start items,” *J. Comput. Sci.*, vol. 29, pp. 9–18, Nov. 2018.

[5] R. Nawaz, P. Thompson, and S. Ananiadou, “Identification of manner in bio-events,” in *Proc. LREC*, 2012, pp. 3505–3510.

[6] H. Qadir, O. Khalid, M. U. S. Khan, A. U. R. Khan, and R. Nawaz, “An optimal ride sharing recommendation framework for carpooling services,” *IEEE Access*, vol. 6, pp. 62296–62313, 2018.

[7] M. Z. Asghar, A. Sattar, A. Khan, A. Ali, F. M. Kundu, and S. Ahmad, “Creating sentiment lexicon for sentiment analysis in Urdu: The case of a resource-poor language,” *Expert Syst.*, vol. 36, no. 3, Jun. 2019, Art. no. e12397.

[8] A. Z. Syed, M. Aslam, and A. M. Martinez-Enriquez, “Lexicon based sentiment analysis of Urdu text using sentiunits,” in *Proc. Mexican Int. Conf. Artif. Intell.*, Berlin, Germany: Springer, 2010, pp. 32–43.

[9] M. Ijaz and S. Hussain, “Corpus based Urdu lexicon development,” in *Proc. Conf. Lang. Technol. (CLF)*, Peshawar, Pakistan, vol. 73, 2007, pp. 1–12.

[10] W. Anwar, X. Wang, and X.-L. Wang, “A survey of automatic urdu language processing,” in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2006, pp. 4489–4494.

[11] A. Daud, W. Khan, and D. Che, “Urdu language processing: A survey,” *Artif. Intell. Rev.*, vol. 47, no. 3, pp. 279–311, Mar. 2017.

[12] S. Kiritchenko, S. Mohammad, and M. Salameh, “SemEval-2016 task 7: Determining sentiment intensity of english and arabic phrases,” in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 42–51.

[13] J. Villena-Román, J. García-Morera, and J. C. González-Cristóbal, “DAEDALUS at SemEval-2014 task 9: Comparing approaches for sentiment analysis in Twitter,” in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 218–222.

[14] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, “SemEval-2016 task 4: Sentiment analysis in Twitter,” 2019, arXiv:1912.01973. [Online]. Available: https://arxiv.org/abs/1912.01973

[15] H. Jang, M. Kim, and H. Shin, “KOSAC: A full-fledged Korean sentiment analysis corpus,” in *Proc. 27th Pacific Asia Conf. Lang., Inf., Comput. (PACLIC)*, 2013, pp. 366–373.

[16] L.-S. Chen, C.-H. Liu, and H.-J. Chiu, “A neural network based approach for sentiment classification in the blogosphere,” *J. Informetrics*, vol. 5, no. 2, pp. 313–322, Apr. 2011.

[17] A. F. Wicaksono, C. Vania, B. Distiawan, and M. Adriani, “Automatically building a corpus for sentiment analysis on Indonesian tweets,” in *Proc. 28th Pacific Asia Conf. Lang., Inf., Comput.*, 2014, pp. 185–194.
S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 41, no. 4, pp. 535–561, Apr. 2014.

Z. U. Rehman and J. S. Bajwa, “Lexicon-based sentiment analysis for urdu language,” in Proc. 6th Int. Conf. Innov. Comput. Technol. (INTECH), Aug. 2016, pp. 497–501.

W. Zhao, Z. Guan, L. Chen, X. He, D. Cai, B. Wang, and Q. Wang, “Weakly-supervised deep embedding for product review sentiment analysis,” IEEE Trans. Knowl. Data Eng., vol. 30, no. 1, pp. 185–197, Jan. 2018.

M. Kamkaraghighi and M. Makrehchi, “Content tree word embedding for document representation,” Expert Syst. Appl., vol. 90, pp. 241–249, Dec. 2017.

Z. Hu, J. Hu, W. Ding, and X. Zheng, “Review sentiment analysis based on deep learning,” in Proc. IEEE 12th Int. Conf. e-Bus. Eng., Oct. 2015, pp. 87–94.

S. M. Rezaenia, A. Ghodsi, and R. Rahmani, “Improving the accuracy of pre-trained word embeddings for sentiment analysis,” 2017, arXiv:1711.08609. [Online]. Available: https://arxiv.org/abs/1711.08609

F. Ali, D. Kwak, P. Khan, S. El-Sappagh, A. Ali, S. Ullah, K. H. Kim, and K.-S. Kwak, “Transportation sentiment analysis using word embedding and ontology-based topic modeling,” Knowl.-Based Syst., vol. 174, pp. 27–42, Jun. 2019.

F. Ali, S. El-Sappagh, S. M. R. Islam, A. Ali, M. Attique, M. Imran, and K.-S. Kwak, “An intelligent healthcare monitoring framework using wearable sensors and social networking data,” Future Gener. Comput. Syst., vol. 114, pp. 23–43, Jan. 2021.

F. Ali, A. Ali, M. Imran, R. A. Naqvi, M. H. Siddqui, and K.-S. Kwak, “Traffic accident detection and condition analysis based on social networking data,” Accident Anal. Prevention, vol. 151, Mar. 2021, Art. no. 105973.

N. Mukhtar and M. A. Khan, “Urdu sentiment analysis using supervised machine learning approach,” Int. J. Pattern Recognit. Artif. Intell., vol. 32, no. 2, Feb. 2018, Art. no. 1851001.

M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. Al-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, and V. Hoste, “SemEval-2016 task 5: Aspect based sentiment analysis,” in Proc. Int. Workshop Semantic Web, 2016, pp. 19–30.

H. Masroor, M. Saeed, M. Feroz, K. Ahsan, and K. Islam, “Transtech: Development of a novel translator for Roman Urdu to English,” Helvion, vol. 5, no. 5, May 2019, Art. no. e01780.

A. Rafae, A. Qayyum, M. Moeenaddin, A. Karim, H. Sajjad, and F. Kasirian, “An unsupervised method for discovering lexical variations in roman urdi informal text,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2015, pp. 823–828.

D. Maynard and K. Bontcheva, “Challenges of evaluating sentiment analysis tools on social media,” in Proc. 10th Int. Conf. Lang. Resour. Eval. (LREC), 2016, pp. 1142–1148.

M. Ganapathibhotla and B. Liu, “Mining opinions in comparative sentences,” in Proc. 22nd Int. Conf. Comput. Linguistics (COLING), 2008, pp. 241–248.

E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, “Learning word vectors for 157 languages,” 2018, arXiv:1802.06893. [Online]. Available: https://arxiv.org/abs/1802.06893

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” Trans. Assoc. Comput. Linguistics, vol. 5, pp. 135–146, Dec. 2017.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” 2013, arXiv:1310.4546. [Online]. Available: https://arxiv.org/abs/1310.4546

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” J. Mach. Learn. Res., vol. 12, pp. 2493–2537, Aug. 2011.

N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” 2014, arXiv:1404.2188. [Online]. Available: https://arxiv.org/abs/1404.2188

Y. Kim, “Convolutional neural networks for sentence classification,” Sep. 2014, arXiv:1408.5882v2. [Online]. Available: https://arxiv.org/abs/1408.5882v2

S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

Alexander Gelbukh is currently a Research Professor and the Head of the Natural Language Processing Laboratory, Center for Computing Research, Instituto Politécnico Nacional, Mexico, and a Honorary Professor of Amity University, India. He has authored or coauthored more than 500 publications in computational linguistics, natural language processing, and artificial intelligence, recently with a focus on sentiment analysis and opinion mining. He is a member of the Mexican Academy of Sciences, a Founding Member of the Mexican Academy of Computing, and a National Researcher of Mexico (SNI) at excellence level 3 (highest). He is the Editor-in-Chief, an associate editor, or an editorial board member of more than 20 international journals, and has been the chair or the program committee chair of over 50 international conferences.