JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 07 Issue 02
2024

Research Article
https://doi.org/10.56979/702/2024

Spelling Variation of Roman Urdu Using Machine Learning

Mudasar Ahmed Soomro[1*], Rafia Naz Memon[2], Asghar Ali Chandio[1], Mehwish Leghari[3], and Muhammad Khalid[1]

[1]Department of Information Technology, Quaid-e-Awam University of Engineering, Science & Technology, Nawabshah, 67480, Pakistan.
[2]Department of Software Engineering, Quaid-e-Awam University of Engineering, Science & Technology, Nawabshah, 67480, Pakistan.
[3]Department of Data Science, Quaid-e-Awam University of Engineering, Science & Technology, Nawabshah, 67480, Pakistan.
[*]Corresponding Author: Mudasar Ahmed Soomro. Email: mudasar1717@gmail.com

Received: March 11, 2024 **Accepted:** July 27, 2024 **Published:** September 01, 2024

Abstract: Spelling variations are common in languages without standardized orthography, such as Roman Urdu (RU), where no established criteria exist for spelling. For example, "2mro" is a nonstandard spelling for "tomorrow." In South Asia, Roman Urdu is widely used, especially on social media and in online product reviews, leading to a proliferation of user-generated spellings. This research compiles a dataset of Roman Urdu words (RUWs) with their spelling variations, collecting 5,244 distinct RUWs, each with one to five. To validate this dataset, in this study, we apply six machine learning (ML) classifiers: Support Vector Machine (SVM), Logistic Regression (LR), Decision Tree (DT), Naïve Bayes (NB), K-Nearest Neighbors (KNN), and Random Forest (RF). Among these, the SVM classifier performs better, achieving an accuracy of 99.96%, surpassing all other algorithms.

Keywords: Roman Urdu; Sentiment Analysis of Roman Urdu; Roman Urdu Spelling Variations; Machine Learning.

1. Introduction

Due to widespread internet usage, which includes a rise in social media platforms such as YouTube, Twitter, and Facebook, these platforms are becoming the most collaborative places for users to interact, express their opinions, and produce vast volumes of textual data. Online forums are an excellent resource for businesses and users to learn about the mindsets of the individuals who express their suggestions and thoughts about the given good or bad services [1]. Sentiment analysis (SA) has seen extensive research applying ML and deep learning (DL) techniques, primarily focusing on resource-rich languages like English and other European languages. However, research on resource-poor languages, such as Russian (RU), has been limited. In Pakistan and other South Asian regions, Urdu is the widely spoken language, and individuals find comfort in communicating using Urdu for social content. This highlights the need for more research in SA for resource-poor languages like Urdu, which could offer valuable insights into the sentiments of a vast population [2]. RUW is considered a non-standard language and is widely prevalent on the web, the biggest challenge in RU sentiment analysis is when there are numerous spellings for a single RUW present in a dataset. If we consider a single RUW such as "Khubsurat" (خوبصورت) (Beautiful), it can also be written as "khubsorat", "khubsoorat", and "khoobsurat" [3] [4], which makes the SA problem even more challenging. The issue is that users often create their spellings for RUWs, which has become a significant challenge for SA. Another problem arises when a RUW has multiple meanings, such as the word "Aam" in RU, which can mean both "mango" (آم) and "common" (عام) [5]. This research study was conducted, mainly due to the unavailability of a RU word with spelling variation dataset. The spelling variation of a single word also becomes challenging when used for RU to English translation.

1.2. Contribution of This Research

The contributions of this research are outlined below:

- In this research, we have created a new dataset that was not previously available. This dataset includes Roman Urdu spelling variations with 1 (one) to 5 (five) words with spelling variations. The dataset contains 5244 Roman Urdu words with spelling variations.
- In the research, we utilized 6 (six) machine learning classifiers to examine the data. With the help of the confusion matrix, precision, recall, and F1 score we evaluation these algorithms.

The rest of the paper is divided into the following sections: Section 2 gives a brief description of the related work, and Section 3 defines the dataset developed and used in this study. The selected algorithms that are used in this study are defined in Section. 4 The overall methodology is described in Section 5. Section 6 discusses the results, while Section 7 concludes the paper and suggests future recommendations.

2. Related Work

English and other developed languages have extensive research studies on text classification. Very little work has been done on text classification and SA in RU, which is the third largest language in the world. In Ref. [6], the author uses two binary and tertiary classification experiments to train a CNN with a max-pooling layer on RU that consists of 10,021 sentences taken from various online sources. Due to factors such as high review volume, larger sentence lengths, and a greater number of classes, the model was unable to perform well on tertiary classification. In Ref [7].

This research explores the examination of personal opinions in Roman Urdu writings and seeks to identify the overall sentiment of the text. Main achievements involve the Bilingual Roman Urdu Language Detector (BRULD) and Roman Urdu Spelling Checker (RUSC), which jointly identify languages, fix spellings, and categorize emotions using a sentiment intensity score. The system attains a language identification precision of 97.1% and a total sentiment categorization precision of 94.3%.

In Ref. [3], the authors perform SA on a dataset of 779 RU reviews. Six different datasets were employed for the review and the dataset was divided into testing sets (40%) and training (60%) that were trained with five classifiers using unigram, bigram, and uni-bigram features. In Ref [8], authors do SA on the resource of poor language RU and determine the complexity of RU and its problems. This work was grouped into three feature techniques: character level, word level, and feature union. The results show that feature union and character-level features perform better than comparing word-level features, with spelling variation complexity of RU serving as the primary cause of poor performance.

The author of Ref. [9] constructed the RU e-commerce dataset RUECD, which includes 26K+ user data evaluations for SA with seven ML classifiers and five DML classifiers. The results showed that LR, SVM, and LSTM performed better than the other models, and the author also observed that words are phonetically similar but have different spellings. Ref. [10] created the DML model for usage by the RUSA-19 dataset. The dataset was classified into the DL classifier RCNN, FRCNN, N-gram, and rule-based model there were experiments in two series one of them was a binary classification that contained (positive and negative) second tertiary classification. The major reason of this study was undertaken was to get better results by consuming less computational time (positive, neutral, and negative).

The author's contribution to Ref. [11] The primary goal of this study is to create a new deep learning technique employed by CNN-LSTM for some of the English and RU sentiment valuations based on the textual dataset collected from user-generated social media and apply a variety of word embedding models and to analyses Word2Vec's performance.

In Ref. [12], the author's contributions are made by the study that initially uses the small datasets for transfer learning to improve and proposes CNN with emphasis on RU SA. This study applied several tests to examine the model's sensitivity and choose the appropriate values for the hyper-parameters that would increase classification accuracy.

In Ref. [13], the author constructs a framework that uses empirical data using two RU datasets, RUECD and RUSA-19. The bidirectional LSTM gives context in both directions; however, the model proposed by the researcher emphasizes more useful factors. The results of the binary and ternary classification that use the final dense Soft-Max output layer are being obtained

From the literature review, most of the existing research work has been reported on the comparison of different classifiers with the performance of ML and DL techniques [3] One problem that arises in all datasets is the presence of numerous spelling variations of RUWs, which hinders algorithms from properly classifying them, some of the existing research papers reported that there is an availability of dataset is not in standard with standard word forms and also RU does not have any spelling criteria defined there for verities of one-word spellings make difficult for SA [6] [14].

3. Dataset

This section outlines the methodologies employed in the compilation of the RU words dataset.

- Links identification

Initially, the primary task was to collect RU data from various sites and social platforms where we could gather RU text data. The sites chosen for data collecting were "Facebook", “daraz”, “pakistan.web”, “twitter”, “youtube”, “hamariweb”, "Twitter" and “dramaonline”.

- Data extraction technique

Data retrieval was done in two ways: first, automatically using web scraping tools, which included sites such as "YouTube," "Facebook," and "dramaonline." Secondly, in cases where this tool couldn't be used on certain websites, we manually copied the data, which included sites like "Twitter" and "Pakistan.web."

- RU word with spelling variation collection method

We collected a total of 20,232 RU reviews from various sites and saved them in an Excel file, as shown in Fig. 1. After tokenized these reviews into words, each review was segmented into individual RUW. Then, we manually searched for all RUW.

The manual method was carried out as follows: another Excel file was created where all the RUWs found were stored. Each RUW was placed in a separate row, and if there were spelling variations or alternative spellings for any RUW, they were added in the next column adjacent to that respective word.

When constructing the dataset, we employed six columns in Excel. Columns one through five were designated for storing spelling variations. Due to the dataset's limitation of capturing a maximum of five spelling variations per RUW, any RUW exceeding this threshold had only its top five variations recorded. These columns were named "Var-1, Var-2, Var-3, Var-4, and Var-5". The sixth column, named "Common", contained the word that appeared most frequently in the reviews among all the spelling variations of that particular RUW, So in the case of the RUW "acha" (اچھا), if it appeared as "a6a" 70 times, "axha" 60 times, and "acha" 150 times within the review data, then "acha" (اچھا) with 150 existences, which is the most frequent spelling, would be placed in the "Common" column. This approach ensures that the most commonly used spelling variation is identified and recorded as the representative spelling for that particular word in the dataset. When searching for RUWs, we encountered some words that were English, such as "glass" and "mobile". We did not include these English words in our dataset.

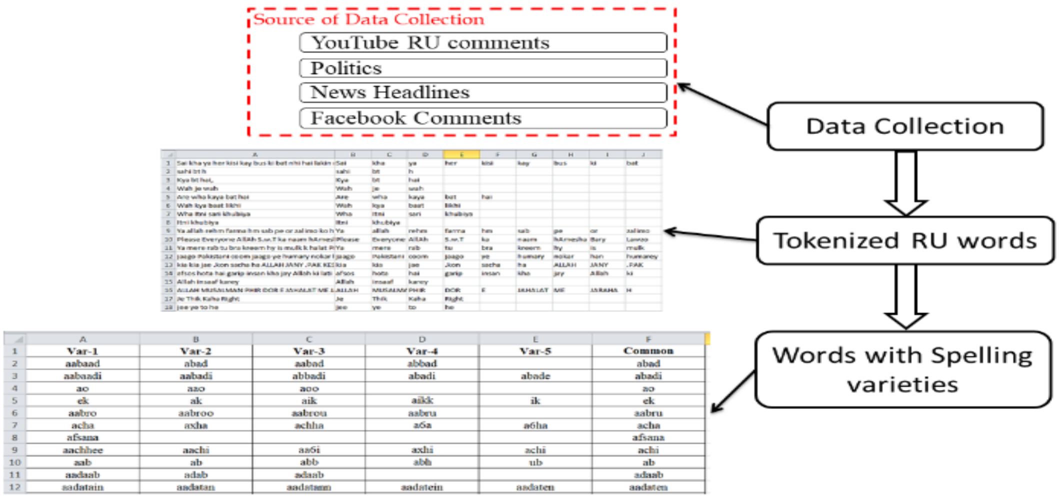

Figure 1. Data Collection and process of creating RU words spelling variation.

Our analysis of 20,232 reviews in Russian RU revealed 5,244 unique words RUWs. These words exhibited spelling variations ranging from one to five. We compiled a dataset encompassing 19,527 RUWs and their associated spelling variations, as detailed in Table 1. This table shows that 1,026 RUWs had five spelling variations, 2,008 RUWs had four spelling variations, 1,987 RUWs had three spelling variations, 181 RUWs had two spelling variations, and 41 RUWs were unique with no spelling variations.

Table 1. Details of the dataset

Spelling Variation	Number of Spelling variation	Total words
Spelling Variation words with 5 (Five) Spellings	1026	1026 X 5 = 5130
Spelling Variation words with 4 (Four) Spellings	2008	2008 X 4 = 8032
Spelling Variation words with 3 (Three) Spellings	1987	1987 X 3 = 5961
Spelling Variation words with 2 (Two) Spellings	181	181 X 2 = 362
Spelling Variation words with 1 (One) Spellings	42	42 X 1 = 42
Total Number of words	**Words 5244**	**Words 19527**

4. Selected algorithms

In this section, we have described the ML algorithms that have been utilized in this research study, namely SVM, LR, RF, KNN, NB, and DT classifiers.

4.1. Naïve Bayes

The text classification using NB produces a class "c*" with the maximum possibility to be a specific document "d" i.e. c* = argmax c p(c|d).

The following is how the Bayes rule can be used to increase this probability:

$$c^* = argmax_c\, p(d) = argmax_c \frac{p(d|c)p(c)}{p(d)} \quad (1)$$

In Eqs.1 the terms "p(d)" stands for "proof," p(c) for "prior probability," and p(d|c) for "possibility" [15].

4.2. Logistic Regression

The most popular and effective algorithm for classification problems isLR [16]. Equations 2, employed for the chosen hypothesis function, consistently predict output values between 0 and 1.

$$0 \leq h_\theta(\theta^T x) \leq 1 \quad (2)$$

The representation of the hypothesis by sigmoid function calculated using Eqs. 3:

$$h_\theta(\theta^T x) = \frac{1}{1+e^{-\theta^T x}} \quad (3)$$

Whenever placed in LR, x is the input variable, and Tx is the hypothesis function is parameterized by h.

4.3. Support Vector Machine

The SVM classifiers work by identifying an optimal hyperplane that maximizes the margin between different classes in the training data. This hyperplane acts as a decision boundary, allowing for the classification of new data points. By maximizing the margin, SVM aims to minimize the risk of misclassification and enhance the model's generalization capabilities. [17]. The classified margin is defined as the distance for each class between the hyperplane and the nearest data points. The SVM's hypothesis-generating function was calculated using Eqs. 4:

$$h_\theta(x) = \{1 \quad if \quad \theta^T x \geq 1 \;\; 0 \quad if \quad \theta^T x \leq -1$$

(4)

4.4. K-Nearest Neighbors

The KNN classifier is a widely used and adaptable ML approach, appreciated for its straightforwardness and user-friendliness. Unlike many other techniques, KNN does not rely on assumptions regarding the data's underlying distribution, which allows it to be a suitable option for the variation of datasets in both classification and regression situations [18]. It effectively manages both numerical and categorical data. KNN works by predicting results based on the similarity between data points in a data set and identifying the KNN to a specific data point by a distance calculation, such as Euclidean distance. The classification or value of a data point is determined by the majority or average of its neighbors. This method exhibits reduced sensitivity to outliers compared to alternative algorithms and can adjust to various patterns by concentrating on the local data structure [19].

4.5. Decision Tree

The root node of a DT is chosen based on the feature that provides the highest information gain, which is calculated for each feature in the dataset [20]. In the context of this, the information gained for the remaining features is obtained, the features for the next level of the tree are selected, and the tree is constantly built using the same technique [21].

4.6. Random Forest

The RF method generates a large number of DT to create a "forest" that is trained using bootstrap aggregation, also known as bagging. Bagging is a technique that enhances the accuracy of machine learning algorithms by combining multiple models [22]. In RF classification, the final prediction is made by combining the predictions from all the trees in the forest. This model takes into justification the majority of votes or the weight of these trees to determine the final result [23] [27].

5. Methodology

In this section, we have defined the methodology in which we have utilized all the steps of ML, enabling us to classify RUWs with spelling variations. Initially, we collected RU reviews and manually separated each word from the reviews. For each RUW, we use a manual search to find its spelling variations, ranging from one to five variations per word. We then compiled this information into an Excel file to create a dataset, as explained in detail in Section 3. The dataset comprises six columns, with the first five columns representing RUWs along with their spelling variations. The final column showcases the most prevalent spelling variation, chosen from the variations encountered most frequently throughout the data collection process.

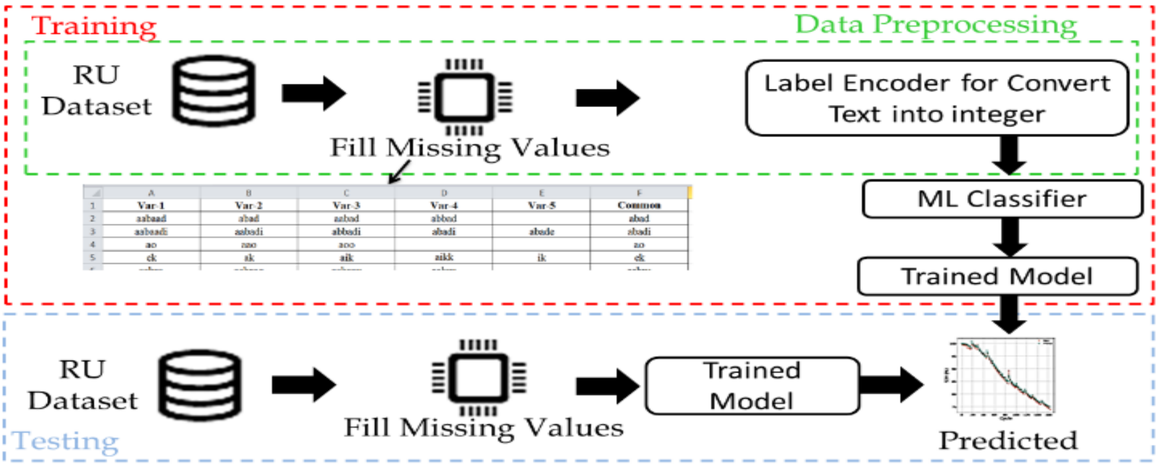

Figure 2. Step-by-step methodology of Machine Learning to implement Roman Urdu words with spelling variation

In the dataset, when a word had four or fewer spelling variations, we left empty columns for the remaining variations. We faced a problem when applying the dataset to the model because the model did not fully accept the dataset due to the presence of many empty columns. To resolve this issue, we applied the missing value method, where we filled the empty columns with (0), as explained in Fig. 2. We split the dataset into data "x" containing "Var-1, Var-2, Var-3, Var-4, and Var-5" as features, and the "common" column was designated as the class label "y". When the preprocessing of the model was complete, we proceeded to train and test the model on the dataset. To achieve this, we ran 15 iterations of each experiment. In this machine learning process, several key settings significantly influence the model's efficacy and reliability. These settings, often referred to as hyperparameters, require careful tuning to achieve optimal performance. The dataset is partitioned into two subsets: 80% for training the model and 20% for evaluating its performance on unseen data [7] [24]. To ensure consistent results across experiments, the data is split using a deterministic rule, guaranteeing that the same data points are assigned to the same training, validation, and testing sets for every run. Additionally, we maintain a balanced representation of classes in both the training and test sets, which is crucial when class sizes are significantly disparate. In machine learning models, textual labels within a dataset are often transformed into numerical representations. This process, known as label encoding, enables the model to effectively interpret and process the data. The model is then trained to learn from the data, looking for patterns using a straightforward approach. These settings work together to create a strong and dependable ML model. Each classification algorithm produced different accuracy values, and we evaluated the performance of all algorithms based on accuracy, precision, recall, and F1-score. We utilized an open-source ML package called scikit-learn [25] [28] for training and classification purposes.

5.1. Evaluation metrics

An accuracy is the measurement that relates to the number of samples to meet the specific correctly identified with the total number of inputs for any established model. Accuracy is often defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

The term "true positive" (TP) in Eq. 5 refers to the number of positive inputs that the system correctly identifies. Similarly, "true negative" (TN) refers to the system accurately identifying negative inputs. "False positive" (FP) occurs when the system incorrectly classifies a negative input as positive, while "false negative" (FN) happens when it mistakenly identifies a positive input as negative. The precision of the spell-checking system reflects its effectiveness and is calculated as the percentage of correctly identified positive cases out of all the positive predictions made by the system [24] [29].

Precision measures the relevance of the information retrieved by the system, indicating how accurate the predictions are. Precision is calculated using Eq. 6, where TP represents true positives and FP represents false positives.

$$Precision\ (P) = \frac{TP}{TP + FP} \quad (6)$$

Eq. 7 can also be used to calculate the precision. If the text document that contains N words and Ci is the right replacement for the word mistakes and Pi is the predicted replacement for the ith word, then precision:

$$Precision\ (P) = \frac{\sum_{i=1}^{N} |C_i \cap P_i|}{\sum_{i=1}^{N} P_i} \quad (7)$$

The recall measure checks the model's completeness. It provides information on the languages handled by spell-checkers. It is the model's selection of detections divided by the total number of accurate detections.

The model will perform better if the recall value is greater. Eqs. 8 and 9 can be used to obtain it.

$$Recall\ (R) = \frac{TP}{TP + FN} \quad (8)$$

$$Recall\ (R) = \frac{\sum_{i=1}^{N} |C_i \cap P_i|}{\sum_{i=1}^{N} C_i} \quad (9)$$

In Eqs. 7 and 9, | Ci ∩ Pi | provides the correct prediction for the incorrect word prediction, Pi represents the total number of predicted words, while Ci indicates the overall number of correctly predicted words. Another evaluation parameter is the "f-measure," which is determined as the harmonic mean of recall and accuracy with equal weights for each [25] [30]. This makes it possible to include of both accuracies and recall in a single score, allowing for the comparison of models and the evaluation of a model's performance. The model's f-measure is calculated using equation 10.

$$f - Measuere = \frac{2PR}{P + R} \quad (10)$$

Here, precision and recall are represented by P and R, respectively.

6. Results and discussion

The experimental results defined in this section provide performance metrics, including Precision, Recall, F1-score, and Accuracy, for all algorithms employed in the study.

Table 2. Classification results (%)

Classifier	Precision	Recall	F1 score	Accuracy
SVM	99.96	99.94	99.94	99.96
DT	98.90	98.50	98.40	98.80
NB	97.60	97.30	97.40	97.50
KNN	93.00	90.00	91.00	92.00

LR	49.83	46.23	47.10	49.84
RF	40.35	32.39	34.17	42.89

As shown in Table 2, the results show that SVM has achieved the highest accuracy of 99.96%. For this specific application, the Sigmoid Kernel validated superior performance, emerging as the optimal choice among available kernel functions. We observed a significant decline in accuracy when employing alternative kernel modes. On the second number, DT achieved an accuracy of 98.80%, followed by NB with 97.50%. KNN achieved 92%, while LR and RF had notably lower accuracies of 49.84% and 42.89% respectively, which are considerably lower. Moreover, when considering Precision, Recall, and F1 scores, the results exhibit subtle deviations from the accuracy figures.

Table 3. Some correctly classified RU words with different spelling varieties

Classifier	Actual input Word			Predicted by Classifier		
	Word-1	**Word-2**	**Word-3**	**Word-1 Prediction**	**Word-2 Prediction**	**Word-3 Prediction**
SVM	Axha	Acha	A6a	Acha (اچھا)	Acha (اچھا)	Acha (اچھا)
KNN	Jhoota	Jhuta	Jhota	Jhuta (جھوٹا)	Jhuta (جھوٹا)	Jhuta (جھوٹا)
DT	Jwb	Jawab	jwab	Jawab (جواب)	Jawab (جواب)	Jawab (جواب)

In Table 3, we have mentioned some RUWs for which we obtained perfect accuracy when we inputted these words into the classifiers and determined whether the output matched the labeled data or not. In this table, "actual input word" refers to the word that we inputted, while "predicted by classifier" refers to the words that the classifier provided as results or output. For instance, in the SVM classifier, when we input the word "Acha (اچھا)", The system successfully identified the correct spelling of the target word from the three provided variations of the RU input. This demonstrates the system's ability to recognize and process different spellings of the same word, indicating robust linguistic processing capabilities. Similarly, we inputted some RUWs such as "Jhuta (جھوٹا)" and "Jawab (جواب)" into KNN and DT classifiers, each with different spellings, and obtained perfectly accurate outputs, just as we had set the outputs to be.

Table 4. Some incorrectly classified RU words with different spelling varieties

Classifier	Actual input Word		Predicted by Classifier		Actual Word Available in Dataset
	Word-1	**Word-2**	**Word-1 Prediction**	**Word-2 Prediction**	
LR	daawat	dawat	Dant (دانت)	Dant (دانت)	Dawat (دعوت)
RF	chuti	chutti	Chunti (چیونٹی)	Chunti (چیونٹی)	Chuti (چھٹی)

In Table 4, we mentioned classifiers with lower accuracies, and the words provided that resulted in incorrect predictions. For example, in the LR classifier, when we searched for the word "Dawat (دعوت)", we inputted two spellings: "Daawat" and "dawat", and received the outputs "Dant (دانت)" and "Dant (دانت)", which were incorrect predictions. The correct output word should have been "Dawat (دعوت)". Similarly, we applied words to the RF classifier that didn't yield correct outputs.

7. Conclusion

In this research, we created a completely new dataset, which was not previously available anywhere. To compile this dataset, we explored various websites to find RU reviews, from which we extracted RUWs and saved them in a CSV file to create the dataset. We then checked the validity of this dataset using six ML classifiers, among which the SVM classifier with Sigmoid Kernel achieved the highest accuracy of

99.96%. This dataset can prove beneficial for the research community because one of the significant challenges within RU is the absence of a standard spelling convention. Due to this, during SA or classification tasks, encountering problems is common. With the help of this dataset, we can establish standard spelling conventions for RUWs, which can significantly improve classification accuracy in SA tasks.

References

1. K. Mehmood, D. Essam, K. Shafi, and M. K. Malik, “Discriminative feature spamming technique for roman urdu sentiment analysis,” IEEE Access, vol. 7, pp. 47991–48002, 2019.
2. H. Ashraf, “The ambivalent role of urdu and english in multilingual pak-istan: a bourdieusian study,” Language policy, vol. 22, no. 1, pp. 25–48, 2023.
3. K. Mehmood, D. Essam, and K. Shafi, “Sentiment analysis system for roman urdu,” in Intelligent Computing: Proceedings of the 2018 Computing Conference, Volume 1, pp. 29–42, Springer, 2019.
4. M. A. Aish, M. Mubeen, M. U. Iqbal, and A. Qaseem, “Sentiment analysis of roman urdu reviews,” UMT Artif. Intell. Rev, vol. 2, no. 1, pp. 00–00, 2022
5. Z. Sharf and S. U. Rahman, “Performing natural language processing on roman urdu datasets,” International Journal of Computer Science and Network Security, vol. 18, no. 1, pp. 141–148, 2018.
6. Z. Mahmood, I. Safder, R. M. A. Nawab, F. Bukhari, R. Nawaz, A. S. Alfakeeh, N. R. Aljohani, and S.-U. Hassan, “Deep sentiments in roman urdu text using recurrent convolutional neural network model,” Information Processing & Management, vol. 57, no. 4, p. 102233, 2020.
7. K. Jawad, M. Ahmad, M. Alvi, and M. B. Alvi, “Rusas: Roman urdu sentiment analysis system.,” Computers, Materials & Continua, vol. 79, no. 1, 2024.
8. R. Joshi, P. Goel, and R. Joshi, “Deep learning for hindi text classification: A comparison,” in Intelligent Human Computer Interaction: 11th International Conference, IHCI 2019, Allahabad, India, December 12–14, 2019, Proceedings 11, pp. 94–101, Springer, 2020.
9. K. Mehmood, On Multi-domain Sentence Level Sentiment Analysis for Roman Urdu. PhD thesis, UNSW Sydney, 2021.
10. B. Chandio, A. Shaikh, M. Bakhtyar, M. Alrizq, J. Baber, A. Sulaiman, A. Rajab, and W. Noor, “Sentiment analysis of roman urdu on ecommerce reviews using machine learning,” CMES-Comput. Model. Eng. Sci, vol. 131, pp. 1263–1287, 2022.
11. A. A. Nagra, K. Alissa, T. M. Ghazal, S. Kukunuru, M. M. Asif, and M. Fawad, “Deep sentiments analysis for roman urdu dataset using faster recurrent convolutional neural network model,” Applied Artificial Intelligence, vol. 36, no. 1, p. 2123094, 2022.
12. L. Khan, A. Amjad, K. M. Afaq, and H.-T. Chang, “Deep sentiment analysis using cnn-lstm architecture of english and roman urdu text shared in social media,” Applied Sciences, vol. 12, no. 5, p. 2694, 2022.
13. D. Li, K. Ahmed, Z. Zheng, S. A. H. Mohsan, M. H. Alsharif, M. Hadjouni, M. M. Jamjoom, and S. M. Mostafa, “Roman urdu sentiment analysis using transfer learning,” Applied Sciences, vol. 12, no. 20, p. 10344, 2022.
14. M. Chhajro, M. Khuhro, K. Kumar, A. Wagan, A. Umrani, and A. Laghari, “Multi-text clas,” 2020.
15. B. A. Chandio, A. S. Imran, M. Bakhtyar, S. M. Daudpota, and J. Baber, “Attention-based ru-bilstm sentiment analysis model for roman urdu,” Applied Sciences, vol. 12, no. 7, p. 3641, 2022.
16. F.-J. Yang, “An implementation of naive bayes classifier,” in 2018 International conference on computational science and computational intelligence (CSCI), pp. 301–306, IEEE, 2018.
17. L. Dong, J. Wesseloo, Y. Potvin, and X. Li, “Discrimination of mine seismic events and blasts using the fisher classifier, naive bayesian classifier and logistic regression,” Rock Mechanics and Rock Engineering, vol. 49, pp. 183–211, 2016.
18. A. S. Manek, P. D. Shenoy, and M. C. Mohan, “Aspect term extraction for sentiment analysis in large movie reviews using gini index feature selection method and svm classifier,” World wide web, vol. 20, pp. 135– 154, 2017.
19. F. F. Xu, U. Alon, and G. Neubig, “Why do nearest neighbor lan-guage models work?,” in International Conference on Machine Learning, pp. 38325–38341, PMLR, 2023.
20. H. Yigit, “A weighting approach for knn classifier,” in 2013 international conference on electronics, computer and computation (ICECCO), pp. 228–231, IEEE, 2013.
21. A. Ekramipooya, M. Boroushaki, and D. Rashtchian, “Application of nat-ural language processing and machine learning in prediction of deviations in the hazop study worksheet: A comparison of classifiers,” Process Safety and Environmental Protection, vol. 176, pp. 65–73, 2023.
22. V. Digumarthi, T. Amin, S. Kanu, J. Mathew, B. Edwards, L. A. Peterson, M. E. Lundy, and K. E. Hegarty, “Preoperative prediction model for risk of readmission after total joint replacement surgery: a random forest approach leveraging nlp and unfairness mitigation for improved patient care and cost- effectiveness,” Journal of Orthopaedic Surgery and Research, vol. 19, no. 1, p. 287, 2024.

23. H. Moradian, M. A. Lau, A. Miki, E. D. Klonsky, and A. L. Chapman, “Identifying suicide ideation in mental health application posts: A random forest algorithm,” Death Studies, vol. 47, no. 9, pp. 1044–1052, 2023.
24. O. Rainio, J. Teuho, and R. Kl´en, “Evaluation metrics and statistical tests for machine learning,” Scientific Reports, vol. 14, no. 1, p. 6086, 2024.
25. E. Kenyeres, A. Kummer, and J. Abonyi, “Machine learning classifier-based metrics can evaluate the efficiency of separation systems,” Entropy, vol. 26, no. 7, p. 571, 2024.
26. Li, J., Xu, X., Guan, Y., Imran, A., Liu, B., Zhang, L., Yang, J.-J., Wang, Q., & Xie, L. (2018). Automatic cataract diagnosis by image-based interpretability. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 3964-3969). IEEE.
27. Karam, M. W., Imran, A., Khan, A. H., & Idrees, S. (2023). Navigating Emotions Across Borders: Deep Learning-Driven Location-Informed Sentiment Analysis of Twitter. Lahore Garrison University Research Journal of Computer Science and Information Technology, 7(3).
28. Ejaz, F., Tanveer, F., Shoukat, F., Fatima, N., & Ahmad, A. (2024). Effectiveness of routine physical therapy with or without home-based intensive bimanual training on clinical outcomes in cerebral palsy children: a randomised controlled trial. *Physiotherapy Quarterly*, *32*(1), 78-83.
29. Latif, J., Xiao, C., Imran, A., & Tu, S. (2019). Medical imaging using machine learning and deep learning algorithms: A review. In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1-5). IEEE.
30. Khan, M. F., Iftikhar, A., Anwar, H., & Ramay, S. A. (2024). Brain Tumor Segmentation and Classification using Optimized Deep Learning. Journal of Computing & Biomedical Informatics, 7(01), 632-640.