Roman Urdu News Headline Classification Empowered with Machine Learning

Rizwan Ali Naqvi[1], Muhammad Adnan Khan[2, *], Nauman Malik[2], Shazia Saqib[2], Tahir Alyas[2] and Dildar Hussain[3]

Abstract: Roman Urdu has been used for text messaging over the Internet for years especially in Indo-Pak Subcontinent. Persons from the subcontinent may speak the same Urdu language but they might be using different scripts for writing. The communication using the Roman characters, which are used in the script of Urdu language on social media, is now considered the most typical standard of communication in an Indian landmass that makes it an expensive information supply. English Text classification is a solved problem but there have been only a few efforts to examine the rich information supply of Roman Urdu in the past. This is due to the numerous complexities involved in the processing of Roman Urdu data. The complexities associated with Roman Urdu include the non-availability of the tagged corpus, lack of a set of rules, and lack of standardized spellings. A large amount of Roman Urdu news data is available on mainstream news websites and social media websites like Facebook, Twitter but meaningful information can only be extracted if data is in a structured format. We have developed a Roman Urdu news headline classifier, which will help to classify news into relevant categories on which further analysis and modeling can be done. The author of this research aims to develop the Roman Urdu news classifier, which will classify the news into five categories (health, business, technology, sports, international). First, we will develop the news dataset using scraping tools and then after preprocessing, we will compare the results of different machine learning algorithms like Logistic Regression (LR), Multinomial Naïve Bayes (MNB), Long short term memory (LSTM), and Convolutional Neural Network (CNN). After this, we will use a phonetic algorithm to control lexical variation and test news from different websites. The preliminary results suggest that a more accurate classification can be accomplished by monitoring noise inside data and by classifying the news. After applying above mentioned different machine learning algorithms, results have shown that Multinomial Naïve Bayes classifier is giving the best accuracy of 90.17% which is due to the noise lexical variation.

[1] Department of Unmanned Vehicle Engineering, Sejong University, Seoul, 05006, Korea.

[2] Department of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan.

[3] School of Computational Sciences, Korea Institute for Advanced Study, Seoul, 02455, Korea.

[*] Corresponding Author: Muhammad Adnan Khan. Email: madnankhan@lgu.edu.pk.

Received: 23 May 2020; Accepted: 29 June 2020.

Keywords: Roman urdu, news headline classification, long short term memory, recurrent neural network, logistic regression, multinomial naïve Bayes, random forest, k neighbor, gradient boosting classifier.

1 Introduction

The language, Urdu, has been expressed in Urdu font as well as Roman Urdu. The concept of Roman Urdu was introduced by the British, however now it has been adopted by Internet users for text messaging. Roman characters that are used in Urdu language and chatting in Roman Urdu on social media are now considered the most typical standard of communication especially in Pakistan [Bilal, Israr and Shahid (2015)]. Mostly, the news on social media is in Roman Urdu but there is no proper categorization system on these social media websites. So, the categorization of the news headlines into their respective domains would help us in conducting research in the relevant domain. If we manage to categorize the news headlines then we will be able to get insight on different aspects, like which newsgroup is popular in the given region at a given time.

In this paper, an attempt has been made to classify the Roman Urdu news headlines into appropriate newsgroups i.e., sports, technology, business, health, international. The main issue we faced at the start of the research was the lack of data. There was no proper and updated sources or dataset of Roman Urdu news, which we could have used directly to train our models and get the required results. There are some websites [Irfan (2020); Imran (2020); Ali (2020); Urdu.92newshd (2019)] which have updated the news in roman Urdu but they comprise of a very small collection of news. So, we scraped the data from Urdu news websites, which have plenty of data and transliterated the Urdu headlines to Roman Urdu headlines [ijunoon (2020)]. The data was further preprocessed by removing the stop words, numerical digits, and different symbols from headlines. After this, we did extensive experimentation with the number of different models including deep learning models which gave us promising results.

The rest of the paper is organized as follows: Section 2 provides a detailed literature review. The data set used has been described in Section 3. Section 4 describes the methodology in which the complete process has been discussed. Discussion and results are presented in Section 5. The conclusion has been discussed in section 6.

2 Literature review

Text classification is the most active research area in natural language processing but there have been only a few efforts to examine roman Urdu due to non-availability of required resources for the Urdu language.

Three machine learning classification models namely Naïve Bayes (NB), K-nearest neighbor (KNN) & Decision tree (DT) have been used for opinion mining of Roman Urdu based text data [Bilal, Israr, and Shahid (2015)]. 150 reviews were used in each category (positive, negative). Among these algorithms, NB gave the best results as compared to KNN & DT.

For analyses of the hotel reviews three different classification algorithms like Support vector machine (SVM), NB & LR are used [Rafique, Malik, Nawaz et al. (2019)]. For this task author created a roman Urdu corpus by scraping comments from different websites. SVM outperformed other algorithms with an accuracy of 87.22%.

Usman et al. [Usman, Shafique, Ayub et al. (2016)] performed Urdu news headline classification. The author created an Urdu corpus of 21679 news which contains several classes like sports, business, health, weird, entertainment, and culture. They applied multiple machine learning algorithms and achieved 94% accuracy using max voting.

Ghulam et al. [Ghulam, Zeng, Li et al. (2019)] have applied supervised classification algorithms of machine learning for news classification documents for predefined categories. They created the data set comprising of 12319 Roman news. Once tokenization has been done, stemming and stop word removal, random forest, Linear Support Vector Machine (LSVM), Bernoulli Naïve Bayes (BNB), Multinomial Naïve Bayes (MNB), Linear Stochastic Gradient Descent (LSGD), and Max Voting (MV) were applied. Max Voting outperformed other machine learning algorithms with a promising accuracy of 94%.

Research on sentiment analysis is increasing day by day but this research is limited to the English language mostly. There are many problems related to sentiment analysis based on the Urdu language so [Khattak, Asghar, Saeed et al. (2020)] focused on the most important problems of sentiment analysis in the Urdu language like text pre-processing, lexical resources and sentiment classification.

Several attempts have been made regarding Roman Urdu sentiment analysis. [Mahmood, Safder, Nawab et al. (2020)] have proposed a model based on deep learning which is a prominent and workable approach and has the capability of performing the featuring engineering itself automatically on a large scale of data corpus efficiently which results in better performance as compared to earlier approaches.

Liu et al. [Liu, Gu, Wang et al. (2019)] have made a successful attempt to create real and natural Chinese handwritten characters with an inherently efficient mechanism for enhancing the quality of the image generated. The observers cannot judge if they are written by a person.

Chen et al. [Chen, Xiong, Xu et al. (2019)] suggested an online incremental learning algorithm derived from the Variable support vector machine (VSVM). The suggested VSVM algorithm can pre-calculated the results and does not lose samples for re-learning results. Thus it increases the efficiency of the algorithm which saves the complexity of calculation time taken by numerical experiments.

Most of the sentimental analysis has been accomplished is either English or Chinese languages but not in Urdu. Bibi et al. [Bibi, Qamar, Ansar et al. (2019)] worked on Urdu news tweets for sentimental analysis. The decision tree algorithm is used for the classification of the Urdu news tweets.

Liu et al. [Liu, Yang, Lv et al. (2019)] have worked on the classification model of Chinese questions and have improved the performance of several classification models. The attention-based BiGRU-CNN model is used to increase the overall performance. By using the methodology suggested, they get the highest score in precision, recall, and F1 score.

Sharf et al. [Sharf and Rahman (2018)] created discourse parser for Roman Urdu. A lexicon-based approach is also used for sentence-level sentiment analysis [Hashim and Khan (2016)]. In 2015, natural language processing techniques are used to find the polarity of a review in the Roman Urdu Opinion Mining System (RUOMiS) [Hashim and Khan (2016)].

CNN is widely known for recognizing the internal structure of the Two-dimensional (2D) structure like image. CNN can also be used to exploit a One-dimensional (1D) structure such as sentences. So, it can work in the field of text analysis and categorization. All the vocabulary words are denoted by one-hot encoding and consider each word as a pixel. So, for example, we define the region of 2 and stride of 1, then the document D="I love you" will return the vectors "I love" and "love you". Now, these text regions are converted into low dimensional feature regions by CNN. This type of setting is called seq-CNN. But if we select the large region size then the weight vector would also become large. So, this increases the complexity of the network. Then another method proposed is to create bag-of-words representation which reduces the region size. This method is called bow-CNN.

Zhang et al. [Zhang, Zhao and LeCun (2015)] proposed a character level CNN method. In this method, the characters are transformed into a one-hot encoding vector. A combination of character-level CNN and RNN used multiple layers of CNN followed by RNN [Xiao and Cho (2016)]. CNN reduces the sequence vectors because it deals with character level modeling. A document usually contains hundreds of thousands of characters, so, feeding it directly to RNN would not be feasible. Extracted features are fed into the RNN that is upright for learning. Finally, there exists a classification layer which calculates the probabilities and classifies the documents.

Zhang et al. [Zhang, Wang, Lu et al. (2019)] proposed such network solution which can be used to deploy CNN for the classification of traffic sign having lower resource setting with good accuracy. They proposed a GLRC model in which CNN with the combination of linguistic resources can get better performance by fully utilizing linguistic resources in real life for the benefit of Aspect based sentiment analysis (ABSA). Many attempts have been made for higher accuracy, robustness, and protection for the security of secret information. Luo et al. [Luo, Qin, Xiang et al. (2019)] have introduced a deep learning-based model for real-time data hiding in an image based on image block-matching and dense convolutional network. It gives more accurate results because CNN extracts the high-level semantic features as compared to low-level features.

The most challenging part of computer vision is strong and accurate visual tracking. Zhang et al. [Zhang, Jin, Sun et al. (2018)] proposed a powerful and accurate target tracking model based on spatial and semantic convolutional features that are extracted from convolutional neural networks in progressive tracking of an object. This algorithm attains a state of the art effects on attribute-based evaluation, qualitative, and quantitative analysis.

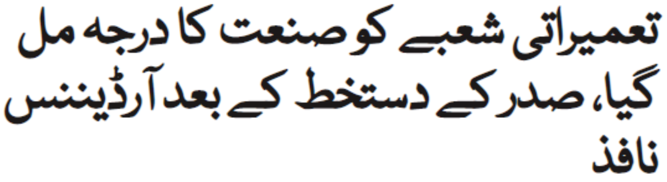
3 Data set

Roman Urdu is a low resource language. Although it is widely used in Pakistan and India for informal communication on the internet and a large amount of unstructured data is available on social media websites like Facebook, Twitter, and chat rooms, but structured/labeled data is not available. To overcome this problem, we first identified the sources of Urdu news websites. Selected websites for data collection were urdupoint.com [Irfan (2020)], geo.tv [Imran (2020)], bolnews.com [Ali (2020)] and 92newshd.tv [urdu.92newshd (2019)].

We selected the five most common categories of news (international, health, sports, business, and technology) and scraped 7351 news of those categories. This scraped news was scripted in the Urdu language. We transliterated them to Roman Urdu using a transliterator created by ijunoon.com [ijunoon (2020)].

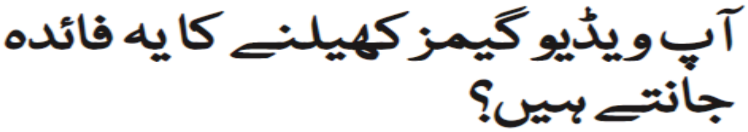
We created two data sets. One with lexical variation was created on real-world data. We gave people Urdu news and asked them to transliterate them to English. There are no standard spellings in Roman Urdu so this dataset was with a lexical variation. The second data set was transliterated through ijunoon transliterator [ijunoon (2020)].

However, there are some Urdu samples given below: where Fig. 1 represents the business category, Fig. 2 represents the health category and Fig. 3 indicates the sports category.



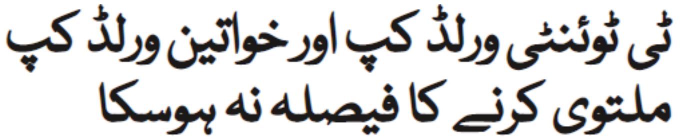
تعمیراتی شعبے کو صنعت کا درجہ مل
گیا، صدر کے دستخط کے بعد آرڈیننس
نافذ

Figure 1: Business category



آپ ویڈیو گیمز کھیلنے کا یہ فائدہ
جانتے ہیں؟

Figure 2: Health category



ٹی ٹوئنٹی ورلڈ کپ اور خواتین ورلڈ کپ
ملتوی کرنے کا فیصلہ نہ ہوسکا

Figure 3: Sports category

4 Proposed methodology

Due to the non-availability of tagged or structured Roman Urdu news data, Urdu news websites were identified which contained the news of selected categories. After that, using scraper we scraped 7351 Urdu news of 5 categories (International, Health, Sports, Business, Technology). Then we transliterated the news to Roman Urdu. For this, we used Urdu to the Roman Urdu transliterator created by ijunoon.com [ijunoon (2020)]. After collecting the data, we removed the stop words, numbers, and punctuation as these do not play any role in news category prediction and are considered noise. The main challenge in Roman Urdu text analysis is lexical variation. Because of no standard spellings, there are a lot of variations in spellings. We created real-world test data with a lexical variation. For that, we gave 600 Urdu news to people and asked them to write them in roman Urdu. We created two sets of test data. One with lexical variation and one without lexical variation. For lexical variation test data, we introduced some rules to control the lexical variation. After analysis of Roman Urdu data, we concluded that vowels do not play much role in the sound of the word. The people use Roman Urdu differently in their writing. So, we removed the vowels (a, e, i, o, u) from the news. Also, some alphabets are used alternatively in Roman Urdu e.g (c, k, q, y, v) so we mapped these alphabets to one.

Two studies were carried out on different test and training data. The first study considered the transliterated test and training data. This data is transliterated using a rule-based transliterator and contains consistent spelling throughout. For the second study, we used real-world test data. This data set had a lexical variation, to control the lexical variation we have used the rule-based approach explained above. For this study, we have used word-level features. We trained 8 (6 traditional, 2 Deep learning) classification models and tested the performance of the classifier. The results are discussed in the next section.

4.1 Features used

We have used the word-level features like Feature and term frequency-inverse document (TFIDF). It's a good feature to use when you must compare the belonging of the word to a particular category. TFIDF also handles common words, which exist among all categories.

$$\mathrm{tfidf}\,(\mathrm{t},\mathrm{d},\mathrm{D}) = \mathrm{tf}(\mathrm{t},\mathrm{d}) \times \mathrm{idf}(\mathrm{t},\mathrm{D}) \qquad (1)$$

where t, d, D represents the terms, document& the collection of documents respectively.

4.2 Algorithms

The following algorithms were used to create a Roman Urdu news headline classifier.

- Multinomial Naive Bayes (alpha=1)
- Logistic Regression (C=1, max-iter=100)
- K-Neighbor Classifier (n_neighnor-5)
- Decision Tree Classifier (default parameters)
- Random Forest Classifier (default parameters)
- Gradient Boosting Classifier (learning rate=0.1)
- Perceptron Based (Multilayer perceptron, one hidden layer, 235 hidden units)
- LSTM

4.3 Deep learning models

For deep learning purposes, the preprocessing is a little different. We find the average length of each sentence and make all the sentences equal to this average. In some cases, it would lose some parts of some sentences and in some cases, the sentence length would be smaller than the average. So, to make all the sentences equal in length we pad the smaller sentences. After tokenizing the text and converting it into the sequence the data is ready for the neural network.

4.3.1 LSTM

LSTM networks is model of RNN which is useful for processing, prediction and classification purposes. The output of last step from RNN is used as an input in current step which tackles the problem long term memory of RNN and produces more accurate predictions.

Input layer: The input layer transforms the short text into a matrix of embedding. The size of the input layer depends upon the number of inputs of that classification model. To equal the size, we must add the additional feature's quantity with a word vector whose dimensions are 32.

LSTM layer: The LSTM layer is connected to the embedding layer. Each LSTM unit is getting two inputs, one from the embedding layer and the other from the previous LSTM unit (hidden state) which helps keep track of the sequence.

Hidden layer: The LSTM layer is connected to the embedding layer. Each LSTM unit is getting two inputs, one from the embedding layer and the other from the previous LSTM unit (hidden state) which helps keep track of the sequence.

Output layer: Then there is another dense layer with 5 nodes equal to the number of newsgroups. The activation function we use is softmax, which is for multi-class classification.

4.3.2 Convolutional neural network (CNN)

CNN is an artificial neural network which uses perceptrons for supervised learning or data analyzation. It is used for image recognition, classification and segmentation purposes. It has input, output, hidden and convolutional layers which are described below.

Input layer: The input layer transforms the short text into a matrix of embedding. The size of the input layer depends upon the number of inputs of that classification model. To equal the size, we have to add the additional feature's quantity with a word vector whose dimensions are 256.

Convolutional layer: 1D convolution is applied to the feature vectors to get the most prominent features in the sequence. This also helps in reducing the size of features on which the network must be trained. Fewer features mean less parameter for training and less complexity.

Hidden layer: The dense layer is connected to the output of the CNN layer to help improve the parameter learning.

Output layer: It is the classification layer that calculates the probabilities and classifies the text based on higher probability.

5 Data analysis and simulation

The dataset is partitioned into two parts training and testing from which training contains 80% data samples and testing contains 20% data samples. Total dataset instances used are 7351 out of which 6751 instances were used during training and 600 instances were used in the testing phase.

We have used 3 measures like Precision, Recall, and F-measure to evaluate the best accuracy of each machine learning algorithm with/without controlled variations.

$$\text{Precision} = \frac{\text{Number of correct positive predictions}}{\text{Total number of positive predictions}} \tag{2}$$

where precision is the division of retrieved news which are co-related.

$$\text{Recall} = \frac{\text{Number of correct positive predictions}}{\text{Total number of positive examples}} \tag{3}$$

where Recall is the fraction of related news that is retrieved.

$$\text{F} - \text{Measure} = \frac{(2 \times \text{Precision} \times \text{Recall})}{(\text{Precision } + \text{Recall })} \tag{4}$$

It represents both Precision and Recall. It uses harmonic mean instead of Arithmetic mean.

Tab. 1 shows the average accuracies of 10-Fold cross-validation of proposed RUNHS-ML models. it observed that Multinomial Naïve Bayes is giving the best accuracy of 90.17% as compared to other algorithms. The accuracy of data with variation is less, which is due to the noise of lexical variation.

Table 1: Average accuracies of proposed RUNHC-ML models

Models	Test data without variation	Real-world test data	Real-world test data with controlled variation
Multinomial naïve bayes (MLB)	90.17	87.25	87.97
Logistic regression (LR)	90.02	88.45	88.54
K nearest neighbor (KNN)classifier	85.75	79.58	81.10
Decision tree (DT) classifier	74.64	63.74	75.76
Random forest (RF) classifier	80.27	72.51	81.87
Gradient boosting (GB)Classifier	83.61	77.29	82.25
Artificial neural network (ANN)	89.83	84.56	86.25
LSTM	88.7	85.68	86.80
CNN	87.6	82.43	83.4

5.1 Multinomial naïve bayes

This classification is part of supervised learning algorithms which is common for the classification of text. The average accuracy of Multinomial Naïve Bayes without variation during testing is 0.90171 and during training is 0.8924. And the accuracy with controlled variation during testing is 0.8798 and during training is 0.8853.

The average accuracy of Multinomial Naïve Bayes (MNB) with 10-Fold cross-validation on our normal data set is 0.8929 as shown in Tab. 2. While accuracy on test data with controlled lexical variation is 0.8797 as shown in Tab. 3.

Table 2: Proposed RUNHC-ML (Multinomial Naïve Bayes) without variation

Class	Precision	Recall	F-Measure
Business	0.89	0.94	0.92
Health	0.88	0.91	0.90
International	0.89	0.86	0.88
Sports	0.94	0.96	0.95
Technology	0.91	0.78	0.84

Table 3: Proposed RUNHC-ML (Multinomial Naïve Bayes) with a controlled variation

Class	Precision	Recall	F-Measure
Business	0.87	0.86	0.86
Health	0.89	0.91	0.90
International	0.81	0.92	0.86
Sports	0.98	0.96	0.92
Technology	0.95	0.73	0.83

5.2 Multiclass logistic regression

Such a classification model which is used to simplify Logistic Regression (LR) into multiclass problems.

The average accuracy of Multiclass logistic regression (MLR) without variation during testing is 0.8996 and during training is 0.8946. And the accuracy with controlled variation during testing is 0.8893 and during training is 0.8859.

The average accuracy of this classification model with 10-Fold cross-validation on our normal data set is 0.8911 as shown in Tab. 4. While accuracy on test data with controlled lexical variation is 0.8854 as shown in Tab 5.

Table 4: Proposed RUNHC-ML (Multinomial logistic regression) without variation

Class	Precision	Recall	F-Measure
Business	0.92	0.91	0.91
Health	0.85	0.91	0.88
International	0.88	0.88	0.88
Sports	0.95	0.94	0.95
Technology	0.89	0.83	0.86

Table 5: Proposed RUNHC-ML (Multinomial logistic regression) with a controlled variation

Class	Precision	Recall	F-Measure
Business	0.93	084	0.89
Health	0.87	0.91	0.89
International	0.83	0.92	0.87
Sports	0.92	0.95	0.94
Technology	0.92	0.79	0.85

5.3 K neighbor classifier

This classification model is known as the prominent classifier so far as it is simple and produces high accuracy.

The average accuracy of K neighbor classifier without variation during testing is 0.8575 and during training is 0.8463. And the accuracy with controlled variation during testing is 0.8111 and during training is 0.8313.

The average accuracy of this classifier with 10-Fold cross-validation on our normal data set is 0.8575 as shown in Tab. 6. While the accuracy of test data with controlled lexical variation is 0.8110 as shown in Tab. 7.

Table 6: Proposed RUNHC-ML (K neighbor classifier) without variation

Class	Precision	Recall	F-Measure
Business	0.84	0.93	0.84
Health	0.83	0.86	0.88
International	0.87	0.78	0.82
Sports	0.89	0.92	0.90
Technology	0.86	0.77	0.81

Table 7: Proposed RUNHC-ML (K neighbor classifier) with a controlled variation

Class	Precision	Recall	F-Measure
Business	0.76	0.84	0.80
Health	0.77	0.84	0.80
International	0.84	0.78	0.80
Sports	0.87	0.89	0.88
Technology	0.86	0.70	0.77

5.4 Decision tree classifier

This is a well-known classification model. A decision tree is a flow chart like structure where each node represents an attribute value. This model has a structure like a flow chart where the attribute value is presented by every node.

The average accuracy of the Decision Tree (DT) classifier without variation during testing is 0.7407 and during training is 0.7188. And the accuracy with controlled variation during testing is 0.7405 and during training is 0.7068.

The average accuracy of the Decision Tree with 10-Fold cross-validation on our normal data set is 0.6374 as shown in Tab. 8. While accuracy on test data with controlled lexical variation is 0.7652 as shown in Tab. 9.

Table 8: Proposed RUNHC-ML (Decision tree classifier) without variation

Class	Precision	Recall	F-Measure
Business	0.81	0.81	0.81
Health	0.64	0.77	0.70
International	0.74	0.64	0.69
Sports	0.82	0.81	0.81
Technology	0.71	0.66	0.68

Table 9: Proposed RUNHC-ML (Decision tree classifier) with a controlled variation

Class	Precision	Recall	F-Measure
Business	0.77	0.82	0.79
Health	0.75	0.80	0.77
International	0.75	0.72	0.74
Sports	0.79	0.81	0.89
Technology	0.78	0.67	0.72

5.5 Random forest classifier

Such a model is useful for making decision trees and different trees inside the forest if we select a random sample from our training set with the help of tree bagging and random subspace technique. Every tree which we made gives a classification. Afterward, we can select the output of related trees from the forest.

The average accuracy of the Random forest (RF) classifier without variation during testing is 0.8625 and during training is 0.8429. And the accuracy with controlled variation during testing is 0.8531 and during training is 0.8362.

The average accuracy with 10- Fold cross-validation on our normal data set is 0.7251 as shown in Tab. 10. While accuracy on test data with controlled lexical variation is 0.8110 as shown in Tab. 11.

Table 10: Proposed RUNHC-ML (Random forest classifier) without variation

Class	Precision	Recall	F-Measure
Business	0.85	0.86	0.85
Health	0.68	0.87	0.77
International	0.79	0.77	0.78
Sports	0.94	0.82	0.88
Technology	0.85	0.69	0.76

Table 11: Proposed RUNHC-ML (Random forest classifier) with a controlled variation

Class	Precision	Recall	F-Measure
Business	0.83	0.87	0.85
Health	0.75	0.88	0.81
International	0.82	0.82	0.82
Sports	0.86	0.87	0.86
Technology	0.85	0.60	0.71

5.6 Gradient boosting classifier

These classifiers are used to associate several weak learning models to generate a powerful predictive model.

The average accuracy of the Gradient boosting (GB) classifier without variation during testing is 0.8369 and during training is 0.8339. And the accuracy with controlled variation during testing is 0.8225 and during training is 0.8268.

The average accuracy of Gradient boosting with 10- Fold cross-validation on our normal data set is 0.7729 as shown in Tab. 12. While accuracy on test data with controlled lexical variation is 0.8263 as shown in Tab. 13.

Table 12: Proposed RUNHC-ML (GB classifier) without variation

Class	Precision	Recall	F-Measure
Business	0.87	0.87	0.87
Health	0.66	0.91	0.77
International	0.89	0.76	0.82
Sports	0.96	0.86	0.91
Technology	0.98	0.76	0.82

Table 13: Proposed RUNHC-ML (GB classifier) with a controlled variation

Class	Precision	Recall	F-Measure
Business	0.83	0.87	0.85
Health	0.75	0.88	0.81
International	0.82	0.82	0.82
Sports	0.86	0.87	0.86
Technology	0.85	0.60	0.71

Tab. 14 shown the comparison of the proposed methodology with previously published articles approaches

Table 14: Comparison table with the previously published article's accuracies with our proposed methodology

Paper Name	Classifiers	Accuracies
Urdu/Hindi News Headline, Text Classification by Using Different Machine Learning Algorithms [Hassan and Zaidi (2019)]	Random Forest Classifier, Gaussian NB, K-Neighbors, Nearest Centroids, Logistic Regression, Multinomial NB, Passive Aggressive Classifier, Perceptron, SGD, Linear SVC, Ridge Classifier	85.51 (best one via Ridge Classifier)
Urdu Text Classification: A comparative study using machine learning techniques [Rasheed, Gupta, Banka et al. (2018)]	Support Vector Machine (SVM), Decision Tree, KNN	68.73 (best one via SVM)
Sentiment Analysis for Roman Urdu [Rafique, Malik, Nawaz et al. (2019)]	NB, LRSGD, SVM	87.22 (best one via SVM)
Sentence Level Sentiment Analysis Using Nouns [Hashim and Khan (2016)]	Lexicon-based approach	86.8 (via lexicon-based approach)
Roman Urdu News Headline Classification [Proposed Methodology]	Multinomial Naïve Bayes, Logistic Regression, K Neighbor Classifier, Decision Tree Classifier, Random Forest Classifier, Gradient Boosting Classifier, ANN, LSTM, CNN	90.17 (best via Multinomial naïve Bayes)

Tab. 14 indicates that our proposed methodology, where we use 8 classification models (6 machine learning, 2 deep learning) out of which Multinomial Naïve Bayes (MNB), gives the highest accuracy of 90.17%, which is better than the accuracy of the previously published methodologies.

6 Conclusion and future work

We first developed the Roman Urdu news data set. We have gone through several phases for the creation and improvement of the Roman Urdu news classifier. Different machine learning algorithms were applied on TFIDF out of which Multinomial Naïve Bayes machine learning classifier is giving the best accuracy of 90.17%. Although it requires some steps for preprocessing data to remove noise, rule-based techniques were used to remove the lexical variation. In the future, we will use character-level features for classification. Character level models make sense too because the real challenge of Roman urdu is the spelling variations. They alleviate the vocabulary problems we encounter on the input of our model. In the future, we can use character level, deep learning models, to control the variations, which could give promising results. Deep extreme learning can be used for further improving system accuracy.

Acknowledgment: Thanks to our families & colleagues who supported us morally.

Funding Statement: This work is supported by the KIAS (Research Number: CG076601) and in part by Sejong University Faculty Research Fund.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Ali, T. (2020): A sharp drop in gold prices. https://www.bolnews.com/urdu/business/2020/04/766594/.

Bibi, R.; Qamar, U.; Ansar, M.; Shaheen, A. (2019): Sentiment analysis for urdu news tweets using decision tree. *IEEE 17th International Conference on Software Engineering Research, Management, and Applications,* https://doi.org/10.1109/sera.2019.8886788.

Bilal, M.; Israr, H.; Shahid, M.; Khan, A. (2015): Sentiment classification of roman-urdu opinions using naïve nayesian, decision tree, and knn classification techniques. *Journal of King Saud University-Computer and Information Sciences*, vol. 28, no. 3, pp. 330-344.

Chen, Y. T.; Xiong, J.; Xu, W. H.; Zuo, J. W. (2019): A novel online incremental and decremental learning algorithm based on variable support vector machine. *Cluster Computing,* vol. 22, no. 3, pp. 7435-7445.

Ghulam, H.; Zeng, F.; Li, W.; Xiao, Y. (2019). Deep learning-based sentiment analysis for roman urdu text. *Procedia Computer Science*, vol. 147, pp. 131-135.

Hashim, F.; Khan, M. A. (2016): Sentence level sentiment analysis using urdu nouns. *Department of Computer Science, University of Peshawar, Pakistan*, pp. 101-108.

Hassan, S. M.; Zaidi, A. (2018): Urdu news headline, text classification by using different machine learning algorithms. https://doi.org/10.13140/RG.2.2.12068.83846.

Imran, S. (2020): Younis khan's advice to the players to work hard and never give up. https://urdu.geo.tv/latest/220816.

Irfan, M. (2020): The number of corona patients in Bahrain has risen to 5,409.

https://www.urdupoint.com/international/news-detail/bahrain/manama/live-news-2422465.html.

Ijunoon (2020): Urdu script to roman urdu transliteration.

https://www.ijunoon.com/transliteration/urdu-to-roman.

Khattak, A.; Asghar, M. Z.; Saeed, A.; Hameed, I. A.; Hassan, S. A. et al. (2020): A survey on sentiment analysis in urdu: a resource-poor language. *Egyptian Informatics Journal*, https://doi.org/10.1016/j.eij.2020.04.003.

Liu, J.; Gu, C. K.; Wang, J.; Youn, G.; Kim, J. U. (2019): Multi-scale multi-class conditional generative adversarial network for handwritten character generation. *The Journal of Supercomputing*, vol. 75, no. 4, pp. 1922-1940.

Liu, J.; Yang, Y. H.; Lv, S. Q.; Wang, J.; Chen, H. (2019): Attention-based BiGRU-CNN for chinese question classification. *Journal of Ambient Intelligence and Humanized Computing*, https://doi.org/10.1007/s12652-019-01344-9.

Luo, Y. J.; Qin, J. H.; Xiang, X. Y.; Tan, Y.; Liu, Q. et al. (2020): Coverless real-time image information hiding based on image block-matching and dense convolutional network. *Journal of Real-Time Image Processing,* vol. 17, no. 1, pp. 125-135.

Mahmood, Z.; Safder, I.; Nawab, R. M. A.; Bukhari, F.; Nawaz, R. et al. (2020): Deep sentiments in roman urdu text using recurrent convolutional neural network model. *Information Processing & Management*, vol. 57, no. 4, pp. 1-14.

Rafique, A.; Malik, K.; Nawaz, Z.; Bukhari, F.; Jalbani, A. H. (2019): Sentiment analysis for roman urdu. *Mehran University Research Journal of Engineering and Technology*, vol. 38, no. 2, pp. 463-470.

Rasheed, I.; Gupta, V.; Banka, H.; Kumar, C. (2018): Urdu text classification: a comparative study using machine learning techniques. *Thirteenth International Conference on Digital Information Management*, pp. 274-278.

Sharf, Z.; Rahman, D. S. U. (2018): Performing natural Language processing on roman urdu datasets. *International Journal of Computer Science and Network Security*, vol. 18, no. 1, pp. 141-148.

Urdu.92newshd. (2019): Pakistani student develops software to detect counterfeit goods.

http://urdu.92newshd.tv/.

Usman, M.; Shafique, Z.; Ayub, S.; Malik, K. (2016): Urdu text classification using majority voting. *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 8, pp. 265-273.

Xiao, Y.; Cho, K. (2016): Efficient character-level document classification by combining convolution and recurrent layers. *arXiv preprint:1602.00367v1.*

Zeng, D. J.; Dai, Y.; Li, F.; Wang, J.; Sangaiah, A. K. (2019): Aspect based sentiment analysis by a linguistically regularized cnn with the gated mechanism. *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 5, pp. 3971-3980.

Zhang, J. M.; Jin, X. K.; Sun, J.; Wang, J.; Sangaiah, A. K. (2018): Spatial and semantic convolutional features for robust visual object tracking. *Multimedia Tools and Applications*, pp. 1-21.

Zhang, J. M.; Wang, W.; Lu, C. Q.; Wang, J.; Sangaiah, A. K. (2019): Lightweight deep network for traffic sign classification. *Annals of Telecommunications*, https://doi.org/10.1007/s12243-019-00731-9.

Zhang, X.; Zhao, J.; LeCun, Y. (2015): Character-level convolutional networks for text classification. *In Advances in Neural Information Processing Systems*, pp. 649-657.