Urdu Text Classification using Majority Voting

Muhammad Usman
Punjab University College of Information Technology
University of the Punjab
Lahore, Pakistan

Saba Ayub
Punjab University College of Information Technology
University of the Punjab
Lahore, Pakistan

Zunaira Shafique
Punjab University College of Information Technology
University of the Punjab
Lahore, Pakistan

Kamran Malik
Punjab University College of Information Technology
University of the Punjab
Lahore, Pakistan

***Abstract*—Text classification is a tool to assign the predefined categories to the text documents using supervised machine learning algorithms. It has various practical applications like spam detection, sentiment detection, and detection of a natural language. Based on the idea we applied five well-known classification techniques on Urdu language corpus and assigned a class to the documents using majority voting. The corpus contains 21769 news documents of seven categories (Business, Entertainment, Culture, Health, Sports, and Weird). The algorithms were not able to work directly on the data, so we applied the preprocessing techniques like tokenization, stop words removal and a rule-based stemmer. After preprocessing 93400 features are extracted from the data to apply machine learning algorithms. Furthermore, we achieved up to 94% precision and recall using majority voting.**

Keywords*—*Text Classification; Tokenization; Stemming; Naïve Bayes; SVM; Random Forest; Bernoulli NB; Multinomial NB; SGD; Classifier; Majority Voting

I. Introduction

Urdu is a well-known language in Indo-Pak regions. There are more than 100 million speakers of Urdu around the globe. It is a national language of Pakistan, one of the twenty-three official languages of India and is a 21st most spoken language in the world.

Due to the increasing use of Internet, electronic data is increasing tremendously. Users nowadays are interested in to find the information from large data sets quickly and efficiently. Therefore, Text Classification is vital and has numerous applications, including identification of text genre, filtration of news according to user's interest, recognition of email whether it is spam or not. Text classification can be helpful in article tagging where we want to assign particular category tag to the articles. There is a lot of work still required in the Urdu language in this area.

The purpose of this paper is to address the challenges in Urdu Text Classification and to introduce a method to achieve maximum accuracy using Machine Learning algorithms along with the max voting system. We used five different machine learning techniques to classify news into seven pre-defined classes which are sports, health, business, entertainment, science, culture and weird. Our method contains five primary processes: tokenization, stop words removal, stemming, applying the machine learning algorithms and assign the class to the document by majority voting.

This paper is structured as follows: Section II explains literature review in which we described some of the work previously done regarding text classification. Section III explains the methodology that describes the complete process that we followed. It includes the details of collecting corpus through crawler, tokenization of the collected data, stop words removal, to convert each word into its stem or root form and then the application of classification algorithms on the preprocessed data. The algorithms are Naïve Bayes, Linear SGD, Multinomial Naïve Bayes, Bernoulli Naïve Bayes, Linear SVM and random forest classifier. Section IV explains results and the last section concludes the summary of all of the work.

II. Literature Review

(Duwairi et al. 2009) compared the performance of three different classifiers Naïve Bayes, k-nearest-neighbor (KNN) and distance based classifier to the Arabic language. They selected ten categories and collected 1000 document corpus. Data cleaning is performed by removing punctuation marks, and the stop words and by formatting tags. The documents were classified using above mentioned three algorithms. Results showed that performance of Naïve Bayes classifier outperformed the other two classifiers.

(Ali et al. 2009) applied Text Classification on the Urdu Language. Tokenization is performed to convert the words normalization. Two algorithms, Naïve Bayes and SVM are used to measure the accuracy by eliminating the features like stop words, stemming and normalizing one by one from the corpus. The result showed 71.31 % accuracy on the baseline, 76.79% after eliminating stop words and 70.08% after stemming. SVM algorithm is applied for the lexicon with maximum accuracy in Naïve Bayes. Their accuracy with the baseline is 78.60%.

(S. Dumais et al. 1998) compared effectiveness of five different classification algorithms (Find Similar, Naïve Bayes, Decision Trees, Bayes Nets, and SVM) in term of accuracy and speed of learning and classification are compared. The dataset is divided into 75% and 25% parts which are used as training

set and development set respectively. The classifier is trained on the training set, and its accuracy is calculated using development set. Find similar has minimum learning time as this algorithm does not have any error minimization procedure and SVM is a second fastest method. The classification speed of all algorithms is almost same. The results showed that SVM is a useful and most accurate algorithm for classification purpose.

(Joachims 1998) expressed the multiple reasons to choose SVM as a classifier for his experiments. SVM does not depend on the number of input features. Furthermore, the document vectors used in this algorithm has very few non-zero entries, and SVM can find out a linear boundary in text categorization. They used two different types of datasets with ten predefined categories for the experiments. They compared the results of SVM with four conventional methods which are Naïve Bayes, Rocchio algorithm, C4.5, and KNN. SVM gives 86.4% results.

(Ahmed, Kashif et al. 2016) used only the SVM for text classification of Urdu headlines. Term frequency was computed for each word in the vocabulary, and inverse document frequency was computed after preprocessing on the corpus (normalization, stop words removal and stemming). They applied fixed value threshold for the unseen words on a list of words developed by calculating TF-IDF. Model is experimented with and without using stemming approach and got improved results with stemming with an accuracy increased by 3.5 %.

(Nidhi and Gupta 2012) performed Text Classification for Punjabi news articles; the results are computed using Naïve Bayes classification, ontology-based classification, and hybrid approach algorithms. They selected seven different categories on sports. Processing phases include stop words removal, stemming, punctuation marks and symbols removal. After feature extraction, the algorithms mentioned above were performed and classes assigned. They showed that the hybrid approach has better results over the other two.

(Odeh et al. 2015) purposed a new method for Arabic Text Classification using vector classification. The proposed approach uses a categorized Arabic documents corpus. The words are calculated to determine the documents keywords. The keywords of the training data categories were compared with the test documents keyword to find out the document's category. After testing, the accuracy of the proposed document was 98% in one category, and the other category has 93%.

(Wajeed et al. 2009) performed experiments on a large number of documents. He focused on hierarchical classification and worked on to extract the Lexicons from the data. The documents vectors are built by those lexicons and applied the Machine learning techniques to these vectors.

(Jain et al. 2015) performs text classification for the Punjabi language. Model is trained on Naïve Bayes and performed testing on four categories from news domain. They faced the issues in corpus collection, so the model is trained on limited data, but their model gives the satisfactory results based on four categories.

(Purohit et al. 2015) formed a word set to get probabilities by using Apriori algorithm and Naive Bayes. They used Porter's stemmer algorithm for tokenization and the two classifiers, Parallel Formulation of Decision Tree and Sequential Decision Tree for Text Classification. By applying the algorithms, 75% accuracy is achieved.

(Dalal et al. 2011) used the pre-processing strategies and text classification algorithms like Naïve Bayes, decision trees to compare with Neural Networks. Additionally, some major issues involved in automatic text classification such as dealing with unstructured text, handling a large number of attributes and choice of suitable machine learning techniques are described. It was concluded that the performance of a classifier that based on Neural Networks is comparably better than Naïve Bayesian method.

(E. Han et al. 2001) used weight-adjusted k-Nearest Neighbor (WAKNN) for text classification. In WAKNN, the whole training set is converted into a matrix where each entry shows frequency of a word in a document which is called term frequency (TF). This matrix is normalized to get all values between 0 and 1. They used cosine similarity to find between documents. That takes documents and weights vector as input. These weight vectors are adjusted to get the best output. In the experiment, different algorithms like WAKNN, K-nearest neighbor and C4.5 are used. It is found by these experiments that WAKNN has the best accuracy as compared to other techniques.

(S. Al-Harbi et al. 2008) evaluated performance of two famous text classification algorithms (SVM and C5.0) on Arabic text. First, the words from the documents converted into a vector of features. To reduce input space of vector, Chi-Squared statistics ($\chi 2$) are used. Chi-Squared statistics is applied on documents frequency, and top 30 features regarding class are selected. Data is divided into 70% and 30% parts which are used as the training and test data. The algorithms mentioned above were used for Text Classification the basis of the selected terms. C5.0 (78.42% accuracy) perform better than SVM (68.65% accuracy) in all categories of the data.

(Dennis et al. 2009) worked on the data of MIT newspaper "The Tech". They classified the historically archived data into six categories, as the data was already labeled; they used supervised learning for their experiments. Five hundred articles were selected from each category and randomly divided into training and testing data and applied three different classifier models on the data: Naïve Bayes, Maximum Entropy, and Probabilistic Grammar classification. Naïve Bayes classifier with Multi-Variate Bernoulli feature set gave 77% accuracy in their experiments.

(Maneka, S. and Radha N. 2013) introduced a technique to classify the text using keywords extraction. TF-IDF and WordNet are used to extract keywords. TF-IDF gives the words that can be possible keywords and WordNet is a lexical database for English words, and it calculates the similarity between the provided words. Naïve Bays, Decision tree, and KNN are the algorithms that are used to classify the text. To evaluate the algorithms on training and test data 10 folds cross-validation technique was used. From the results, Naïve Bayes gives the efficient result among all with 0.3 Root Mean Square error.

(Kamruzzaman et al. 2010) proposed a new algorithm that uses association rule mining along with Naïve Bayes. Although, the accuracy of this algorithm is acceptable, but this classification method requires time-consuming steps.

(Nidhi and Gupta 2012) applied text classification on Indian Punjabi language. Pre-processing includes special character removal, stop words removal and stemming. The Corpus is collected contains 150 Punjabi documents, which is extracted from Punjabi news websites and trained model for seven categories. The ontology-based classification algorithm is applied as it did not require training data. The result is 85%.

(Li et al. 2015) performed the experiment by using the text categorization method to predict the trend of Chinese stock. Text process was divided into three steps: Text representation, features selection, and Text categorization. KNN and SVM algorithms are performed for text classification. 1000 Poly Real Estate news are collected for the model. By applying process and techniques, SVM model shows better results with 83% precision.

(Jain et al. 2015) reviewed different techniques proposed by various authors for Punjabi text classification. The techniques used for Punjabi Text Classification are Rocchio's algorithms, K-nearest neighbors, Naïve Bayes, decision tree and neural networks. There is not much work has been done in the Punjabi language. So it is the very challenging task to perform classification on Punjabi data.

(Bhumika et al. 2013) performed research to get what is known about text classification so that it will make easy to decide what next steps should be made. For this purpose, Text Classification process is described which contains documentation collection, pre-processing, indexing, features extraction, classification and then evaluation to get fallout and accuracy. Types of text mining algorithms are text classification, discovering association and clustering algorithm. Further algorithms are discussed in each of the mentioned algorithms, and their advantages and disadvantages are discussed. It will make the decision easy which approach should be followed with which algorithm.

III. METHODOLOGY

Our methodology contains a step-wise procedure; we started from the Urdu language corpus collection and then used some preprocessing techniques for features selection to apply actual classification algorithms. The flow chart in Fig-1 summarizes the process which we followed for our technique.

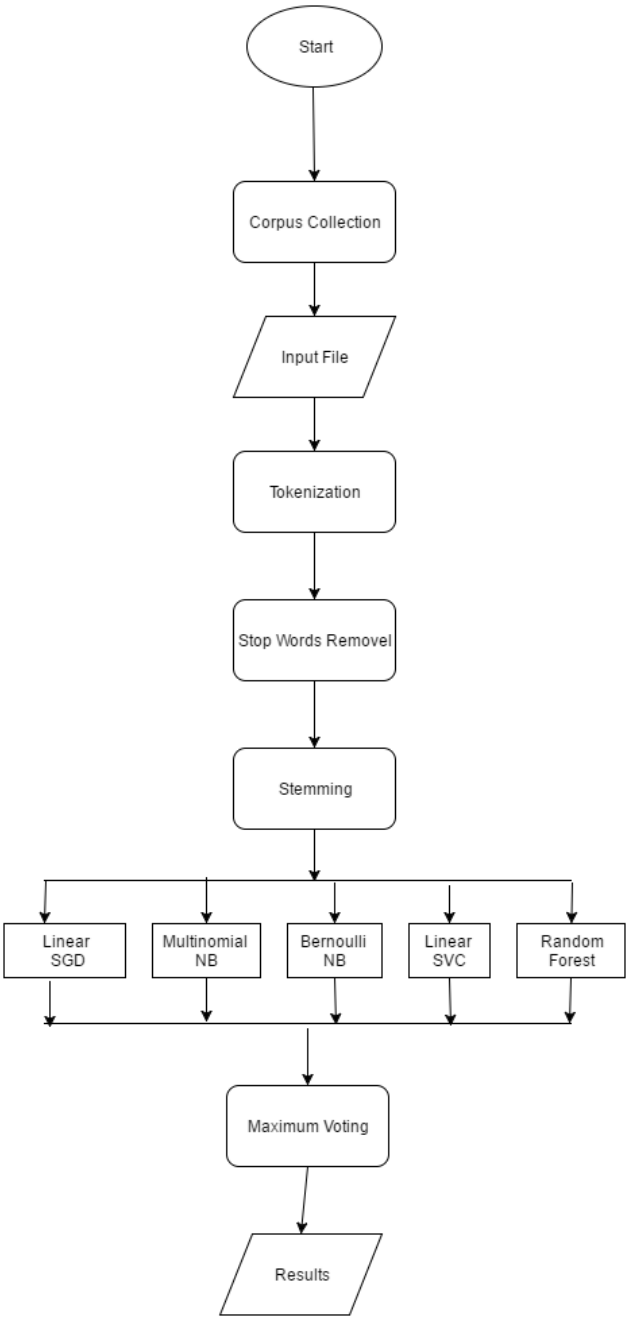

Fig. 1. Flow chart of Urdu News Classification

A. CorpusCollection

Extensive training data plays a vital role in the development of a model that uses supervised learning algorithm. For this purpose, we write multiple crawlers to collect data from different news websites, e.g., express.pk, urdutimes.com, cricnama.com, bbcurdu.com, dawnnews.tv. In total, we collected 21769 documents, and there are more than five million tokens and 120166 vocabularies. Data is collected category-wise in the text files, and categories are as follow: Business, Entertainment, Health, Science, Sports, Culture, and Weird. These categories are the classes used to classify our news data. After data collection, we performed preprocessing techniques like data cleaning, tokenization and stemming to convert the data in a required form on which we need to run algorithms. Details of the collected data are as following:

TABLE I. CORPUS DETAILS

Category	Total Documents	Total Tokens	Total Vocabulary
Sports	5288	1879731	34620
Culture	5285	1142748	48967
Entertainment	4395	652137	33252
Business	2683	560112	23154
Weird	1688	303815	23496
Science	1470	327176	21571
Health	960	212293	15377
Total Dataset	21769	5078012	120166

The sum of total vocabulary in all seven categories is 222008, but the vocabulary of the complete data set is 120166.

B. Tokenization

While processing any natural language, tokenization is often considered as a very first step. Languages usually use white spaces, periods, punctuation marks as a word boundary. In our process, we tokenized the data set into words by space and by removing sentence boundary identifiers (i.e.'', '.', '?', '!').

Example of Tokenization

پاکستان کی سرٹیفیکیشنز کافی بہتر ہیں

The above mention sentence will be tokenized as:

پاکستان	کی	سرٹیفیکیشنز	کافی	بہتر	ہیں

C. Stop Words Removal

The words which are either not useful for the proposed classification models or used as prepositions are included in the stop words list. In our case, we maintained a list of stop words to omit from our text to extract meaningful data for the classifiers. We built the stop words list manually, which also includes the Arabic I'rāb and has more than 1000 entries. We used a look-up based approach to remove the stop words. Some examples of stop words are given below:

پوچھتی	لگیں	گی	انہوں	لوجی
تب	کیوں	تھے	آئے	کھولیں
رکھی	چلو	معلوم	برائے	کہاں
والے	لگی	چلا	کریں	ہوچکے

Example of stop words

The same example which was used in tokenization, a dictionary list is generated after stop words removal is:

پاکستان کی سرٹیفیکیشنز کافی بہتر ہیں

پاکستان	سرٹیفیکیشنز	کافی	بہتر

D. Stemming

In Urdu language word stemming is critical for Information Retrieval (IR). The stem is considered as the base word or a root word. Stemming is an iterative approach to reducing the words into their root form e.g. (منتظمین, Organizers) into (منتظم, Organizer). Urdu stemming rules are entirely different from English, and there is not very much work done in this language. There are many challenges we have faced while stemming. We are using two different approaches in our paper, lookup based and rule based approach. Lookup based method is fast, but it requires a significant amount of memory for words, and rule based approach requires a keen knowledge of literature. After some analysis of literature and study of Urdu grammar, 23 rules are used in this paper to get the stem of an Urdu word. We developed a stemmer which stems words into their base form by using the approaches mentioned above.

Approaches for Stemmer

Following are two approaches for stemming:

1) *Look Up based Approach*
2) *Rule-based Approach*

Look Up based Approach

After extracting stop words, we get the list of words vocabulary. The words in the vocabulary can be in their different forms; the words can be in singular, plural, past tense, or having affixes attached. So we need to get their base form to classify in a particular class. We maintain a dictionary of about 120000 unique Urdu words which are used in look-up approach to validate the word formed after applying stemming rules. We maintain a dictionary of about 120000 manually verified Urdu words which are used in look-up method to validate the word formed after applying stemming rules. For all words in the list, we go through the dictionary and check whether that word exists in the dictionary. If we find the word, we consider it a stemmed word. Our model will always return a legitimate word.

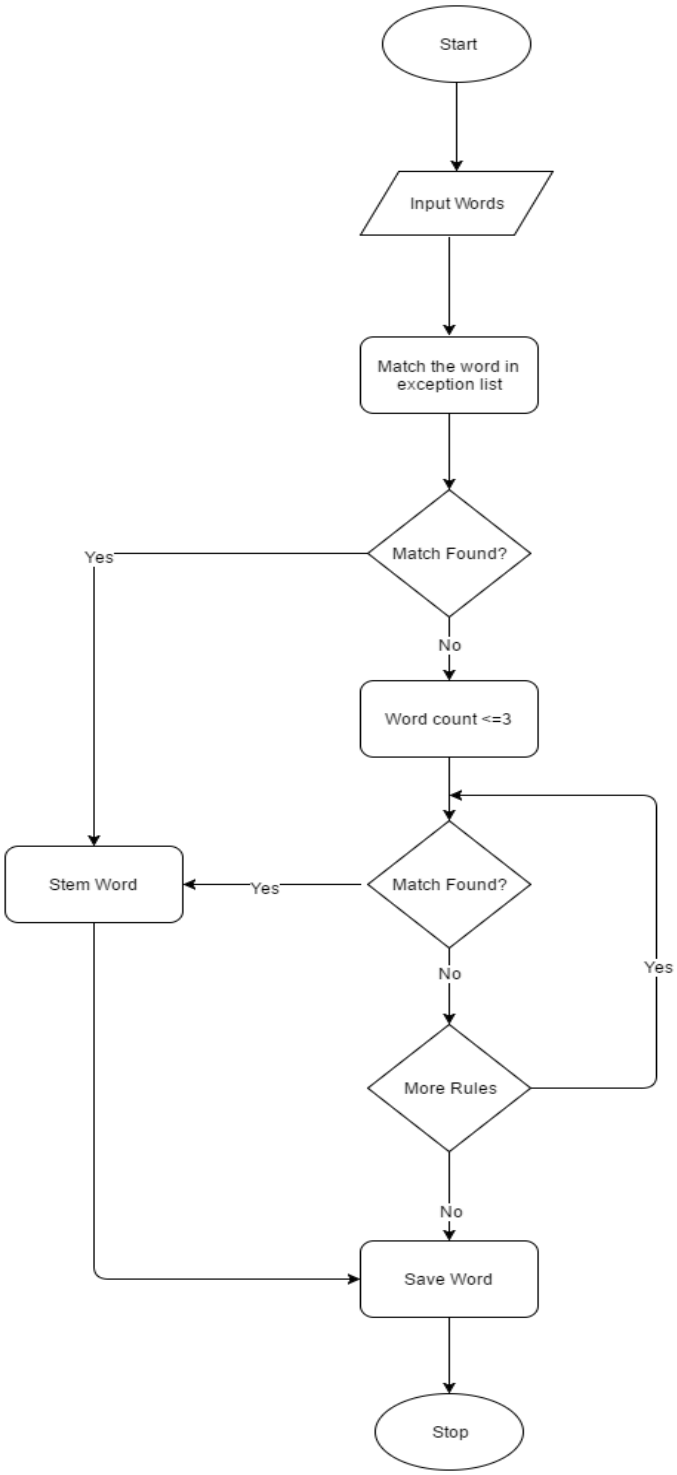

Fig. 2. Flow Chart for Stemmer

Rule-based Approach

In this approach, we implemented 23 rules to convert word tokens into their stem form.

Exception List

Some words cannot stem by the stemming rules, and they are unique words, such words are exceptions. We have an exception list and at the very first step, our algorithm tries to find out the word within the exception list and does not apply any further rule if it founds the word in the list.

1) Length Based Decision

Domain experts suggest that if a word length is less than or equal to three, then the word is already in its root/stem form. So accept that word in the same form.

مات	رات	دین
ابو	چھت	ماں

Ó

2) Affixes Removal

Affixes are the addition to the base form of a word to modify its meaning or create a new word. Affixes are of two types: prefix and suffix.

- The prefix is a morpheme that can either be of a single, two or more than two letters attached at the beginning of a word.

Word	Rule	Stem
بااثر	Remove →با	اثر
پراعتماد	Remove →پر	اعتماد

- The suffix attaches at the end of a word. It may also have a single, two or more than two characters.

Word	Rule	Stem
دغاباز	Remove →باز	دغا
خوددار	Remove →دار	خود

3) Word ends with ے (Rule 1)

If a word ends with ئے (hamza, bari-yay), remove ئے (hamza, bari-yay) and add ا (Alif).

Word	Rule	Stem
چاہئے	ئے→ا	چاہا
جوئے	ئے→ا	جوا

4) Word ends with ے (Rule 2)

If a word ends with ئے (hamza, bari-yay), remove یٔ(hamza,) and add ا (Alif) at the end of the word.

Word	Rule	Stem
گزرگئے	Remove →یٔ Add →ا	گزرگیا
کمائے	Remove →یٔ Add →ا	کمایا

5) Word ends with ے (Rule 3)

If a word ends with ئے (hamza, bari-yay),remove ئے (hamza, bari-yay).

Word	Rule	Stem
چڑھائے	Remove →ئے	چڑھا
اٹھائے	Remove →ئے	اٹھا

6) Word ends with ے (Rule 4)

If a word ends with ے (bari-yay), remove ے (bari-yay and add ا (Alif)at the end of the word.

Word	Rule	Stem
گزرنے	Remove →ے Add →ا	گزرنا

دھرنے	Remove →ے Add →ا	دھرنا

7) Word ends with ے (Rule 5)

If a word ends with ے (bari-yay), remove ے (bari-yay) and add ہ (Hey) at the end of the word.

Word	Rule	Stem
علاقے	Remove →ے Add →ہ	علاقہ
پیمانے	Remove →ے Add →ہ	پیمانہ

8) Word ends with ے (Rule 6)

If a word ends with ے (bari-yay), remove ے (bari-yay).

Word	Rule	Stem
ضلعے	Remove →ے	ضلع
درمیانے	Remove →ے	درمیان

9) Word ends with وں (Rule 7)

If a word ends with وں (wao-non guna), remove وں (wao-non guna).

Word	Rule	Stem
سرخیوں	Remove →وں	سرخی
باغیوں	Remove →وں	باغی

10) Word ends with وں (Rule 8)

If a word ends with وں (wao-non guna), remove وں (wao-non guna) and add ہ (Hey)at the end of the word.

Word	Rule	Stem
حملوں	Remove →وں Add →ہ	حملہ
چھاپوں	Remove →وں Add →ہ	چھاپہ

11) Word ends with وں (Rule 9)

If a word ends with وں (wao-non guna),remove وں (wao-non guna) and add ا (alif) at the end of the word.

Word	Rule	Stem
اندھیروں	Remove →وں Add →ا	اندھیرا
کپڑوں	Remove →وں Add →ا	کپڑا

12) Word ends with یں (Rule 10)

If a word ends with یں (‘yay’ and non-guna) remove یں (‘yay’ and non-guna)

Word	Rule	Stem
خبریں	Remove →یں	خبر
ٹیمیں	Remove →وں	ٹیم

13) Word ends with ؤں (Rule 11)

If a word ends with ؤں(Hamza-wao , non-guna), remove ؤں (Hamza-wao , non-guna).

Word	Rule	Stem
رہنماؤں	Remove →ؤں	رہنما
بندوؤں	Remove →ؤں	بندو

14) Word ends with یاں (Rule 12)

If a word ends with یاں (yay ,alif , non-guna), remove یاں (yay , alif , non-guna) and add ی (choti-yay) at the end.

Word	Rule	Stem
گرفتاریاں	Remove →یاں Add →ی	گرفتاری
کہانیاں	Remove →یاں Add →ی	کہانی

15) Word ends with یات (Rule 13)

If a word ends with یات (yay ,alif , te), remove یا(yay , alif).

Word	Rule	Stem
ضروریات	Remove →یا	ضرورت
سہولیات	Remove →یا	سہولت

16) Word ends with یات (Rule 14)

If a word ends with یات (yay ,alif , te), remove last ا (alif).

Word	Rule	Stem
ہدایات	Remove →ا	ہدایت
شکایات	Remove →ا	شکایت

17) Word ends with یات (Rule 15)

If a word ends with یات (yay ,alif , te), remove یات (yay ,alif , te).

Word	Rule	Stem
جنگلیات	Remove →یات	جنگل
نمکیات	Remove →یات	نمک

18) Word ends with ات (Rule 16)

If a word ends with ات (alif ,te), remove ات(alif ,te).

Word	Rule	Stem
خواہشات	Remove →ات	خواہش
بخارات	Remove →ات	بخارا

19) Word ends with ات (Rule 17)

If a word ends with ات (alif ,te), remove ات(alif ,te) and add ہ (Hey) at the end of the word.

Word	Rule	Stem
جذبات	Remove →ات Add →ہ	جذبہ
مقدمات	Remove →ات Add →ہ	مقدمہ

20) Word ends with ین (Rule 18)

If a word ends with ین (yay, non-guna), remove ین (yay, non-guna).

Word	Rule	Stem
ماہرین	Remove →ین	ماہر
صارفین	Remove →ین	صارف

21) Word ends with ی (Rule 19)

If a word ends with ی (choti-yay), remove ی (choti-yay) from end. If legit then accept.

Word	Rule	Stem
پاکستانی	Remove →ی	پاکستان
غلطی	Remove →ی	غلط

22) Word ends with ی (Rule 20)

If a word ends with ی (choti-yay), replace ی (choti-yay)with ا (Alif).

Word	Rule	Stem
کتنی	Remove →ی Add →ا	کتنا

چاہتی	Remove →ی Add →ا	چاہتا

23)Word ends withوں (Rule 21)

If a word ends withوں(wao-non guna), removeوں(wao-nonguna).

Word	Rule	Stem
کسانوں	Remove →وں	کسان
مرغیوں	Remove →وں	مرغی

24)Word ends withنگ (Rule 22)

If a word ends with نگ (non ,gaf), remove نگ (non , gaf).

Word	Rule	Stem
انجینئرنگ	Remove →نگ	انجینئر
مارکیٹنگ	Remove →نگ	مارکیٹ

25)Word ends withز (Rule 23)

If a word ends with ز (zae), remove ز (zae).

Word	Rule	Stem
فادرز	Remove →ز	فادرر
سسٹرز	Remove →ز	سسٹر

E. Stemmer Accuracy

The following strategy computes the accuracy:

- Split whole corpus in training, development and testing data by 60, 20 and 20 ratios respectively.
- Apply all approaches mentioned in 3.4.2 section on training data and trained our model.
- Using trained model and manually stemmed dataset, calculated accuracy on development data.
- Identify new rules and exceptional words by analysis on stemmed words generated using development data and manually stemmed dataset.
- Add newly identified rules in rules list and new exceptional words in the exceptions list.
- Run updated model on test data and obtained accuracy.
- Repeat all steps three times.

TABLE II. STEMMER ACCURACY TABLE

Iterations	Development Data Accuracy	Test Data Accuracy
First	0.89	0.91
Second	0.93	0.94
Third	0.94	0.95

F. Classification Algorithms

After applying all preprocessing, we have a list of features to apply classification algorithm. Data is divided into two parts: training data and testing data. Out of 21769 documents, 70% of the documents are considered as training dataset and 30% as a testing dataset. Following are the details of the algorithms we have applied. Each classifier gives different accuracy score, and can suggest a different class to a document as compared to the other algorithms. We assigned a class to a document by majority voting from each algorithm. We discuss the detailed results from each classifier in this section.

1) Multinomial Naïve Bayes classifier

Naïve Bayes technique is a set of supervised learning algorithms. Naïve Bayes techniques are very common in text classification.

As name proposes, Multinomial Naïve Bayes works on the data that is distributed among multiple features. We consider vocabulary V as features (N total features) so we can define a document as an occurrence of features in an ordered sequence.

We compute a vector $\theta_y = (\theta_{y1}, \ldots., \theta_{yn})$ for each category$y$. We can calculate the probability of occurrence of each feature i in a category y as P(x_i | y).

So we can estimate the category by the following calculation.

$$\theta = \frac{\alpha + N_{yi}}{\alpha n + N_y}$$

Where $\alpha = 1$ is to add Laplace smoothing for unseen features. The precision of this model is 87%.

TABLE III. MULTINOMIAL NAÏVE BAYES RESULTS

Classes	Precision	Recall	F1-Score
Business	0.979	0.988	0.983
Entertainment	0.978	0.938	0.957
Health	0.924	0.967	0.945
Science	0.659	0.895	0.759
Sports	0.738	0.595	0.659
Weird	0.999	0.978	0.988
Culture	0.800	0.822	0.811
Average / Total	0.868	0.883	0.872

2) Bernoulli Naïve Bayes Classifier

Bernoulli is also similar to Multinomial technique. It also works on the data that is discrete and distributed among N features. The only difference is Multinomial computes the frequency of each feature in a particular category whereas the Bernoulli is more like binary distributed and assign 1/0 if the feature is seen or not in a category.

$$P(x_i|y) = P(I \mid y)\, x_i + (1 - P(I \mid y))\,(1 - x_i)$$

As the above equation shows Bernoulli's is interested only in the occurrence of a word and penalizes if a feature does not be seen in a category. So it gives the better results on small data sets. Once the model is trained, we test it using testing data (McCallum and Nigam 2002). The precision of this model is 84%.

TABLE IV. BERNOULLI NAÏVE BAYES RESULTS

Classes	Precision	Recall	F1-Score
Business	0.949	0.987	0.968
Culture	0.981	0.597	0.742
Entertainment	0.647	0.979	0.779
Health	0.804	0.885	0.842
Science	0.760	0.625	0.686
Sports	1.000	0.935	0.966
Weird	0.742	0.842	0.789
Average / Total	0.840	0.836	0.825

3) Linear SVM

Another algorithm which we are using in our classification system is Linear SVM. In SVM we treat features as 2D space and try to find the closest point which we call support vector

because features are treated as vectors in space, once we find the closest point then draw a line connecting them. We have already made a line that separates these two points as far as possible, and the SVM says the best separated line is, that bisects the two points and is perpendicular to the line that connects them. We are making some connection between documents and classes by connecting them as well as separating them to the particular distance. Whenever a document appears, we map it to a point and check the point on the other end of the separating line, to predict its class. By applying this algorithm, we get the precision up to 89%.

TABLE V. LINEAR SVM RESULTS

Classes	Precision	Recall	F1-Score
Business	0.980	0.993	0.986
Culture	0.962	0.963	0.962
Entertainment	0.955	0.945	0.950
Health	0.739	0.920	0.820
Science	0.785	0.689	0.734
Sports	0.995	0.994	0.994
Weird	0.831	0.808	0.820
Average / Total	0.892	0.902	0.895

4) Random Forest Algorithm

The fourth classification algorithm in our classifier system to get accuracies is Random Forest Classifier. In this model, we make decision trees by selecting a random sample from our training set using tree bagging and random subspace technique. We generate different trees in the forest by choosing random samples. Each tree gives us a classification. Then we choose the output of most correlated trees from the forest.

Once all of the trees assembled in the forest, the labeled data get pass through the trees. Here come the proximities, the proximity of two events get increased by one if both events lie on the same leaf node.

In the end, proximities get normalized with the Total number of trees in the forest. The precision of Random Forest Algorithm on our data set is 83%.

TABLE VI. RANDOM FOREST

Classes	Precision	Recall	F1-Score
Business	0.887	0.995	0.938
Culture	0.847	0.975	0.906
Entertainment	0.926	0.891	0.908
Health	0.645	0.784	0.708
Science	0.675	0.443	0.535
Sports	0.998	0.982	0.990
Weird	0.825	0.542	0.654
Average / Total	0.829	0.802	0.805

5) Linear SGD Classifier

Linear SGD is the simplest algorithm for classification. In this algorithm, we use the gradient descent approach of gradually increasing or decreasing parameters to achieve our goal. With the combination of linear regression, we randomly initialize our parameters and compute accuracy through error function.

In this method, we learn the weights for our data that help to minimize the error of the model. In each cycle, the weights get updated until the error reaches to its minimum threshold. The equation is

$$\omega=\omega-\alpha^* \Delta$$

Where ω is a learned weight, and α is learning rate. The precision of this algorithm is 90%.

TABLE VII. LINEAR SGD RESULTS

Classes	Precision	Recall	F1-Score
Business	0.978	0.991	0.985
Entertainment	0.963	0.962	0.963
Health	0.948	0.946	0.947
Science	0.796	0.913	0.851
Sports	0.790	0.734	0.761
Weird	0.995	0.994	0.995
Culture	0.826	0.798	0.812
Average / Total	0.899	0.906	0.902

6) Max Voting:

The maximum voting technique is quite famous in decision making that is implemented to get best voted predicted class by all the algorithms. For this technique, all of the results generated by above five algorithms is gathered and then take mod of the predicted class of each document. Below table shows the accuracy of the maximum voting technique for every class.

TABLE VIII. MAX VOTING RESULTS

Classes	Precision	Recall	F1-Score
Business	0.980	0.994	0.987
Culture	0.978	0.966	0.972
Entertainment	0.953	0.974	0.963
Health	0.876	0.913	0.894
Science	0.911	0.880	0.895
Sports	0.998	0.987	0.993
Weird	0.898	0.893	0.896
Average / Total	0.942	0.944	0.943

IV. RESULTS

We applied five algorithms on the data and got different accuracies. We also applied some preprocessing techniques like tokenization, stop words removal and stemming before the application of classifiers. Does the preprocessing help to maximize the classifier's accuracy? To check the difference between preprocessed data and raw data we run our algorithms on the tokenized data and data after stemming has been applied. The following table describes the brief summary of all the applied algorithms.

TABLE IX. ACCURACY OF ALGORITHMS BEFORE STEMMING

Algorithms	Precision	Recall	F1-score
Multinomial NB	0.671	0.683	0.638
Bernoulli NB	0.713	0.714	0.711
Linear SVM	0.772	0.795	0.774
Random Forest	0.771	0.772	0.756
Linear SGD	0.763	0.781	0.764
Max Voting	0.825	0.815	0.811

TABLE X. ACCURACY OF ALGORITHMS AFTER STEMMING

Algorithms	Precision	Recall	F1-score
Multinomial NB	0.868	0.883	0.872
Bernoulli NB	0.742	0.842	0.789
Linear SVM	0.892	0.902	0.895
Random Forest	0.825	0.542	0.654
Linear SGD	0.899	0.906	0.902
Max Voting	0.942	0.944	0.943

Clearly, from the table, Maximum Voting technique gives better precision, recall, and f1-score. Regarding the algorithm used, SVM gives us better precision where linear SGD have better recall and f1-score.

The purpose of this experiment is to develop Urdu Text Classifier using the best approach which is used in previous experiments. The main parts of this experiment are to stem data and to classify that data into classes. For stemming we follow two papers; Urdu based stemmer by (Akram et al. 2009) and stemmer for multi Urdu text by (Ali et al. 2016). Our stemming accuracy is 95% which is more than Asma's experiment (91%) and Waheed's experiment (85.02%).

For text classification, we have applied five different algorithms and their accuracies are compared with each other. Best accuracy we find by applying Linear SVM and Linear SGD algorithms on our data set. To get the maximum accuracy Max voting technique is also being implemented in this paper and gives 94% accuracy.

V. Conclusion

The paper presents the work performed to develop a text classifier for Urdu. The process we followed is stepwise, In the first step it tokenizes the data, applies pre-processing techniques including stop words removal and stemming using different algorithms, on the tokenized data. The experimental evaluation using seven different news classes are showing good accuracies by using five different algorithms, and max voting technique. Authors believe that the trained models will also work well on all type of Urdu text data, and their research will be used and help to develop innovative solutions using Urdu text.

VI. Future Work

Urdu Text classification has much room for improvement. Currently, we are using space-based tokenization, we can use the techniques of text segmentations, POS tagging to get better information from data and we can also use lemmatization instead of stemmer to get more improved results of text classification.

References

[1] Duwairi, Rehab, Mohammad Nayef Al-Refai, and Natheer Khasawneh. "Feature reduction techniques for Arabic text categorization." Journal of the American society for information science and technology 60.11 (2009): 2347-2352.

[2] Ali, Abbas Raza, and Maliha Ijaz. "Urdu text classification." Proceedings of the 7th international conference on frontiers of information technology. ACM, 2009..

[3] Dumais, Susan, et al. "Inductive learning algorithms and representations for text categorization." Proceedings of the seventh international conference on Information and knowledge management. ACM, 1998..

[4] Joachims, Thorsten. "Text categorization with support vector machines: Learning with many relevant features." European conference on machine learning. Springer Berlin Heidelberg, 1998.

[5] AHMED, Kashif, et al. "Framework for Urdu News Headlines Classification."Journal of Applied Computer Science & Mathematics 21 (2016).

[6] Nidhi, Vishal Gupta. "Domain Based Classification of Punjabi Text Documents using Ontology and Hybrid Based Approach." 24th International Conference on Computational Linguistics. 2012.

[7] Odeh, Ashraf, et al. "Arabic Text Categorization Algorithm using Vector Evaluation Method." arXiv preprint arXiv:1501.01318 (2015).

[8] Wajeed, Mohammed Abdul, and T. Adilakshmi. "Text classification using machine learning." Journal of Theoretical and Applied Information Technology 7.2 (2009): 119-123.

[9] Jain, Ubeeka, and Kavita Saini. "A Review on the Punjabi Text Classification using Natural Language Processing.", 2015.

[10] Purohit, Anuradha, et al. "Text Classification in Data Mining.", 2015.

[11] Dalal, Mita K., and Mukesh A. Zaveri. "Automatic text classification: a technical review." International Journal of Computer Applications 28.2 (2011): 37-40.

[12] Han, Eui-Hong Sam, George Karypis, and Vipin Kumar. "Text categorization using weight adjusted k-nearest neighbor classification." Pacific-asia conference on knowledge discovery and data mining. Springer Berlin Heidelberg, 2001.

[13] Al-Harbi, S., et al. "Automatic Arabic text classification." (2008).

[14] Ramdass, Dennis, and Shreyes Seshasai. "Document classification for newspaper articles." (2009).

[15] Menaka, S., and N. Radha. "Text classification using keyword extraction technique." International Journal of Advanced Research in Computer Science and Software Engineering 3.12 (2013).

[16] Kamruzzaman, S. M., Farhana Haider, and Ahmed Ryadh Hasan. "Text classification using data mining." arXiv preprint arXiv:1009.4987 (2010).

[17] Nidhi, Vishal Gupta. "Domain Based Classification of Punjabi Text Documents using Ontology and Hybrid Based Approach." 24th International Conference on Computational Linguistics. 2012.

[18] Li, Bozhao, et al. "Text Categorization System for Stock Prediction."International Journal of u-and e-Service, Science and Technology 8.2 (2015): 35-44.

[19] Jain, Ubeeka, and Kavita Saini. "A Review on the Punjabi Text Classification using Natural Language Processing.", 2015.

[20] Bhumika, Prof Sukhjit Singh Sehra, and Prof Anand Nayyar. "A review paper on algorithms used for text classification." International Journal of Application or Innovation in Engineering & Management 3.2 (2013): 90-99.

[21] McCallum, Andrew, and Kamal Nigam. "A comparison of event models for naive bayes text classification." AAAI-98 workshop on learning for text categorization. Vol. 752. 1998.

[22] Akram, Qurat-ul-Ain, Asma Naseer, and Sarmad Hussain. "Assas-Band, an affix-exception-list based Urdu stemmer." Proceedings of the 7th workshop on Asian language resources. Association for Computational Linguistics, 2009.

[23] Ali, Mubashir, et al. "A Rule based Stemming Method for Multilingual Urdu Text." International Journal of Computer Applications 134.8 (2016): 10-18.