On Exhaustive Evaluation of Eager Machine Learning Algorithms for Classification of Hindi Verses

Prafulla B. Bafna[1], Jatinderkumar R. Saini[2]*
Symbiosis Institute of Computer Studies and Research
Symbiosis international (Deemed) University, Pune, India

***Abstract*—Implementing supervised machine learning on the Hindi corpus for classification and prediction of verses is an untouched and useful area. Classifying and predictions benefits many applications like organizing a large corpus, information retrieval and so on. The metalinguistic facility provided by websites makes Hindi as a major language in the digital domain of information technology today. Text classification algorithms along with Natural Language Processing (NLP) facilitates fast, cost-effective, and scalable solution. Performance evaluation of these predictors is a challenging task. To reduce manual efforts and time spent for reading the document, classification of text data is important. In this paper, 697 Hindi poems are classified based on four topics using four eager machine-learning algorithms. In the absence of any other technique, which achieves prediction on Hindi corpus, misclassification error is used and compared to prove the betterment of the technique. Support vector machine performs best amongst all.**

Keywords—Classification; eager machine learning algorithm; Hindi; prediction

I. Introduction

Most of the past and contemporary research works have targeted English corpus document classification and prediction. In online and offline systems, documents are continuously generated, stored, and accessed every day in large volumes. Classifying text according to the contents present helps to produce groups based on tokens present in the text. The maximum work is done in text classifiers focuses on English corpus, but text in Hindi on the web has come of age since the advent of Unicode standards in Indic languages. The Hindi content has been growing by leaps and bounds and is now easily accessible on the web at large. Generally, researchers have focused on Hindi text but only for Natural Language Processing (NLP) activities like word identification, stemming and summarization [1].

Classification or supervised learning groups the labeled data based on the features of data. The data is partitioned as training and testing. Classifiers are broadly divided into eager and slow learners. Eager learners require a long period for training and less time for predicting. For slow learners data gets trained early but it takes more time for a prediction. Eager classifiers give better results than lazy classifiers for text data, so these classifiers are chosen. Naive bayes, Support Vector Machines, Neural Network and Decision tree are popularly used eager classifiers. A decision tree is a classifier, which generates several rules and tables. As a result, rules are placed in the form of decision trees.

Artificial neural network (ANN) has minimum three layers , input, hidden and output. Depending upon the input given and its respective output the network consisting of nodes gets trained. All nodes and layers are interconnected with each other and pass the values generated through the functions, it means that every node present in layer n is connected to various nodes present in tier n-1, inputs connected to respective nodes and nodes present in layer n+1. Output nodes show the classes to which a particular input object belongs.

Classification and regression is carried out through “Support Vector Machine” known as a supervised machine learning algorithm. Each data item is plotted as appoint in n dimensional space. It considers features of the object which are represented by coordinates of a point. SVM differentiates points in different hyperplanes.

Naive Bayes works with text classification. Every unique term is treated as a feature while processing text. Naive Bayes is an eager learner and simple algorithm and termed as strong performer to achieve the classification of text. Naïve byes works best when features are dependent on each other [2].

To apply any classification algorithm on text data first it needs to be converted into structured form. There are several techniques like bag of words, term frequency inverse document frequency and so on, which selects important terms from the corpus based on the frequency of the terms [3].

II. Bakground

In spite of Hindi being used for communication by a large number of people in the world, lots of research work in the field of text classification [4-6] focuses on English. The reason may be processing Hindi corpus is a difficult task.

Topic models are built on Hindi corpus using algorithms, namely Latent Semantic Indexing (LSI), Non-negative Matrix Factorization (NMF), and Latent Dirichlet Allocation (LDA). Many visualizations in the form of trees were used to focus the analysis and results. The outcomes of Hindi text topic modelling gives best results as compared to some outcomes generated on English corpus [7]. To apply any classification techniques the data should be in the tabular form. Various techniques are available to store such types of data for example bag of words [8]. But it creates dimension curse, as all terms in the corpus are considered. High dimensions affect the performance of the algorithm. To reduce high dimensions, only significant words need to be considered. Classification will execute in less time if the top significant words are selected.

*Corresponding Author.

To improve the classification process, the text is preprocessed by removing stop words, etc. [9-10]. Generally (TF-IDF) is a popularly used technique that transforms text data into matrix form. The measure represents the significance of the token with respect to text documents considering the entire corpus. In document processing, it acts as a weighting unit. In spite of increasing word count proportional to the number of documents in which it is present, The TF-IDF ignores the most commonly occurring words, by offsetting count of the words in the entire corpus. [11]. Accuracy, and misclassification errors are used to evaluate classifiers. Hindi is a morphologically rich language. Hindi words have many morphological variants that present the same concept but differ in tense, plurality, etc. A lightweight stemmer is proposed for Hindi, which conflates terms by providing suffix list. The stemmer has been evaluated by computing under stemming and over stemming figures for a corpus of documents [12-14].

Various methods like simulated annealing, genetic algorithms and differential evolution are used which finds out the required solution. Multi-parent mutation and crossover operations are used by the differential evolution algorithm. Results of the methods are input to Naïve Bayes classifiers and its different variations. [15-18]. The proposed algorithm works well in case of text classification as compared with other existing algorithms.

ANN is used to classify the text present in the Arabic language. ANN model is generated for an Arabic corpus. Document representation using different methods along with the feature weights [26] are used and results into identifying important terms. Each Arabic document is represented by the term weighting scheme. The term weighting scheme is used to represent the document. To choose the most significant terms, SVD is used to avoid dimension curse.

Back-propagation neural network (BPNN) and modified back-propagation neural network (MBPNN) are proposed to categorize the text. To avoid dimension curse and to improve the efficiency of algorithm an efficient feature selection method is used.

Training time required for BPNN is slow thus it is modified to enhance the speed required to train. Instead of using a vector space model which is based on term frequency, latent semantic analysis is used. LSA uses only important terms and considered a semantic relationship between the terms and builds concept space. The news dataset is used to prove the efficacy of prosed technique [27].

Different Machine learning algorithms are used to classify the text present in different questions. Two approaches namely Bag-of-words and bag-of-grams are used to construct vector space. Syntactic terms present in the question are identified using a kernel function. Comparative analysis of algorithm performance is being carried out [28] Classification of Hindi text documents includes dividing the documents as training and testing corpus and applying classifiers on the labeled text. Handwritten and printed text documents are partitioned into specific classes. The algorithm is implemented on Hindi text which has Hindi printed and handwritten. The system will be useful for discrimination between handwritten and printed text [19-21].

The text is classified based on emotional features present into it. There are nine categories of emotional features. One category represents one class. Term frequency is used to handle overlapping features. Naïve byes and support vector machines are executed on a set of 55 poems having 10531 words [22-25].

This research is unique because

1) Prediction of Hindi poem using four eager classifiers is achieved.

2) Performance evaluation of the classifier is carried out.

3) Scalability is achieved by processing 697 poems.

III. Research Methodology

The proposed approach initiates with corpus removal of stop words and finds out top N frequent terms using TF-IDF weights on the corpus of poems having three groups. The N value is called a threshold, which is 50 % of maximum TF-IDF weight. Stemming and lemmatization are not used. It effectively removes all unuseful words. Different classifiers are available in the literature, the proposed approach applies all eager classifiers on the term document matrix and the model is built using each classifier. Naïve byes and random forest algorithms are applied. Their performance is evaluated using accuracy. Support vector machine performs best in comparison with remaining algorithms. Fig. 1 depicts the research methodology.

In the paper terms, dimensions, words and tokens are used as synonyms, interchangeably. The paper is organized as follows. The work done by other researchers on the topic is presented as a background in the next section. The third section presents the methodology; the fourth section depicts Results and discussions. The paper ends with a conclusion and future directions. Table I shows steps in the proposed approach.

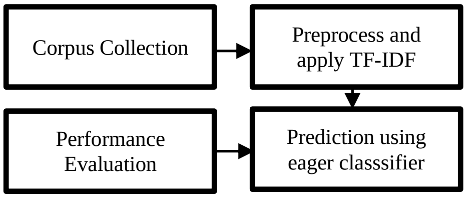

Corpus Collection
Preprocess and apply TF-IDF
Performance Evaluation
Prediction using eager classsifier

Fig. 1. Diagrammatic Representation of Research Methodology.

TABLE. I. Steps and Packages used in the Proposed Approach

Step No	Step	Library/Package/Function
1	Documents are pre-processed and stop words are removed.	library(udpipe)
2	Apply TF-IDF to calculate token weights	dtm_tfidf
3	Select terms having token weights greater than 50 % threshold	dtm <- document_term_matrix(dtm_threshold)
4	Apply and evaluate classifiers	model=naive_bayes(as.factor(type) ~., data=train),
5	Select the best classifier and Predict category of new poem	p1=predict(Naïve_byes,train)

1) Corpus collection and preparation: The proposed approach initiates with data collection and preparation. It includes the process of generating, loading and preprocessing of the corpus. Corpus containing Hindi Text is preprocessed to remove the stop word. It is then partitioned into training and validation sets. The corpus comprises of poems belonging to three categories. The classes or categories are "बाल गीत" ("Bal geet") means children's' poems, "उपदेश गीत" ("Updesh geet") means life lesson teaching poem and "भजन" ("bhajans") means devotional songs. "देश भक्ति" (Desh Bhakti) means patriotic songs. The size of the corpus is 697 and it is downloaded from different websites [29].

2) Converting unstructured data into structured data: Converting unstructured data into the structured one is the next corpus of poems is converted into a vector space model. TF-IDF is used on a set of documents, and token weight is calculated. Terms or tokens having a weight greater than or equal to the threshold are considered. The Document term matrix (DTM) is input to the classifier algorithm. This step selects important tokens present in the corpus and selected significant tokens are further used to form a vector space model.

3) Model training using different classifiers and evaluation: The labeled dataset or corpus is trained based on different values of input and its corresponding output. Eager Classifiers are applied on the DTM. Models are generated and trained using the training corpus. A confusion matrix is found out for all four algorithms and misclassification error was used to evaluate the performance of the algorithm. The best classifier is selected to predict the category of the new poem. Figure specifies the diagrammatic representation of research methodology

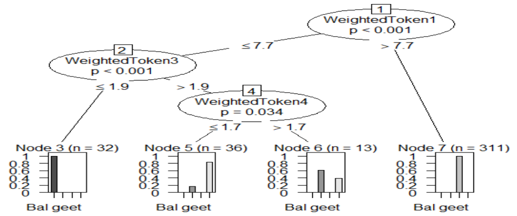
4) Prediction: The best performing classifier is used to predict the category of a poem. It was observed that the support vector machine predicts the class of a poem in a more accurate way.

IV. Results and Discussions

Fig. 2 shows a decision tree for Hindi poems' corpus along with token weights. The corpus of 697 poems is used to build the model. Each token's significance with respect to each category is generated by a decision tree. The figure depicts a particular node represented as "Bal geet" category. The rules based on the weighted tokens for each category are generated.

Fig. 3 shows Naïve bayes classification. The model is a plot for weighted token 4 on the Y axis, it represents a density of Weighted token4 for different categories of poems. The graph clearly represents four different categories of poems namely Bal geet, Bhajan Updesh geet and DeshBhakti geet. and Updesh geet are classified as Bhajans.

Fig. 4 shows the SVM plot. SVM divides the data into two significant hyperplanes. It clearly shows that the upper part of hyperplane consists of poems having category "Desh Bhakti geet". Rest of the poems are distributed in the lower part. Overlapping of poems for two classes can be observed. Confusion matrix depicts the misclassification between Updesh geet and Bhajans.



```
Fitted party:
[1] root
[2] WeightedToken1 <= 7.7
| | [3] WeightedToken3 <= 1.9: Bal geet (n = 32, err = 0.0%)
| | [4] WeightedToken3 > 1.9
| | | [5] WeightedToken4 <= 1.7: Updesh geet (n = 36, err = 16.7%)
| | | [6] WeightedToken4 > 1.7: Bhajan (n = 13, err = 38.5%)
| [7] WeightedToken1 > 7.7: DeshBhakti (n = 311, err = 0.0%)

Number of inner nodes: 3
Number of terminal nodes: 4
```

Fig. 2. Decision Tree and Classifier Rules.

```
Confusion Matrix and Statistics

  Reference
Prediction Bal geet Bhajan DeshBhakti Updesh geet
 Bal geet  11  0  0  0
 Bhajan   0  1  0  3
 DeshBhakti 0  1  85   0
 Updesh geet  0  4  0  4

Overall Statistics

  Accuracy : 0.9266
   95% CI : (0.8605, 0.9678)
 No Information Rate : 0.7798
 P-Value [Acc > NIR] : 3.524e-05

  Kappa: 0.8005
```

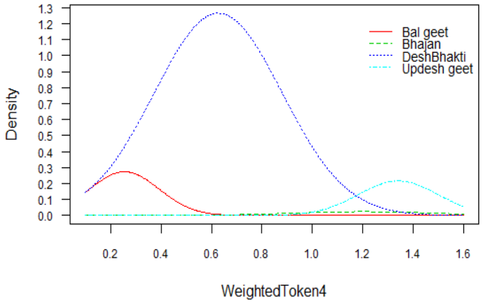


Fig. 3. Model Fitting by Naïve Bayes and Accuracy of Prediction.

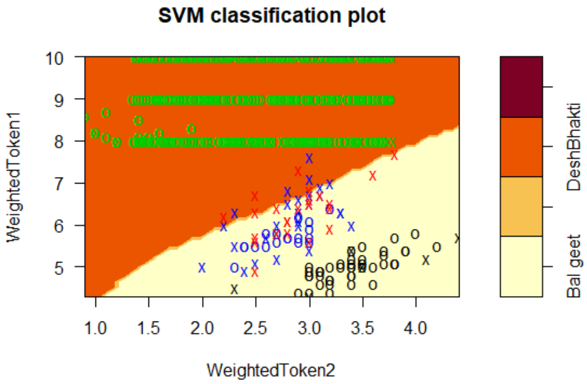


Fig. 4. Hyperplanes Produced by Support Vector Machine.

Fig. 5 shows the confusion matrix along with the Prediction of type of poem carried out using SVM. It is clear that the class accuracy is 0.96, also actual and predicted results are shown that is 11 poems actually belonging to Bhajan class are classified as Updesh geet. All categories of poem can be seen in plot represented by different colours.

Fig. 6 represents the neural network generated for all categories of the poems. Four significant tokens are acting as input to a network. Weights applied by two hidden layers are shown in the figure. The network is trained to identify the tokens most helpful in an accurate classification. These input-weight products are summed and then the sum is passed through a node's activation function. Accuracy of the prediction is calculated comes out to be 0.88 for 500 poems id depicted. Blue coloured lines represent hidden layers.

Table II shows a misclassification error produced by all four algorithms for different corpus size. The support vector machine gives best results for all samples of poems. The error produced by SVM is less for all sample sizes for all four algorithms.

```
Prediction Bal geet Bhajan DeshBhakti Updesh geet

 Bal geet  39 0  0   0

 Bhajan   0 8  0   6

 DeshBhakti  0 0  373   0

 Updesh geet  0  11  0  30


Overall Statistics


   Accuracy : 0.9636

    95% CI : (0.9424, 0.9787)

 No Information Rate : 0.7987

 P-Value [Acc > NIR] : < 2.2e-16


   Kappa: 0.8951
```

Fig. 5. Prediction of Verse type using SVM.

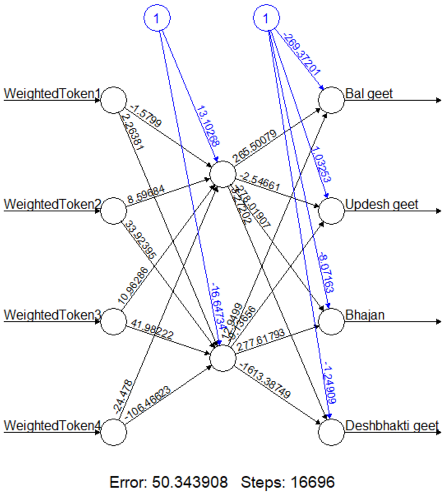


Fig. 6. Neural Network showing Input, Hidden and Output Layer.

TABLE. II. COMPARISON OF MISCLASSIFICATION ERROR PRODUCED BY CLASSIFIERS

Sr. No	Corpus Size	SVM	Decision tree	Neural network	Naïve Byes
1	100	0.45	0.61	50.34	0.51
2	250	0.46	0.63	50.45	0.53
3	400	0.47	0.64	50.82	0.54
4	697	0.47	0.66	50.85	0.55

V. CONCLUSIONS

The current study achieves the prediction of a class of Hindi poem, unlike the other published research works, which have focused on classification of English text. Additionally, the contribution of this study is the exhaustive evaluation of the eager classifiers. The formation of the classes was achieved through the TF-IDF. Government and non-government agencies can use the approach to classify reports, initiatives, different schemes, etc. Experiments are conducted on a corpus of 697 poems. The current work is the first of its kind in the world, which employs prediction and performance evaluation for Hindi corpus comprising of verses.

REFERENCES

[1] Heimerl, F., Lohmann, S., Lange, S., & Ertl, T. (2014, January). Word cloud explorer: Text analytics based on word clouds. In 2014 47th Hawaii International Conference on System Sciences (pp. 1833-1842). IEEE.

[2] Ray, S. K., Ahmad, A., & Kumar, C. A. (2019). Review and Implementation of Topic Modeling in Hindi. Applied Artificial Intelligence, 1-29.

[3] Ramanathan, Ananthakrishnan, and Durgesh D. Rao. "A lightweight stemmer for Hindi." In the Proceedings of EACL. 2003.

[4] Diab, D. M., & El Hindi, K. M. (2017). Using differential evolution for fine tuning naïve Bayesian classifiers and its application for text classification. Applied Soft Computing, 54, 183-199.]

[5] Puri, S., & Singh, S. P. (2018). Hindi Text Document Classification System Using SVM and Fuzzy: A Survey. International Journal of Rough Sets and Data Analysis (IJRSDA), 5(4), 1-31.

[6] Pal, K., & Patel, B. V. (2020). Model for Classification of Poems in the Hindi Language Based on Ras. In Smart Systems and IoT: Innovations in Computing (pp. 655-661). Springer, Singapore.

[7] Saini, J. R., & Desai, A. A. (2011). Identification of Hindi Words Used in Pornographic Unsolicited Bulk E-Mails. IUP Journal of Systems Management, 9(2).

[8] Garg, A., & Saini, J. R. A Systematic and Exhaustive Review of Automatic Abstractive Text Summarization for Hindi Language.

[9] Kaur, J., & Saini, J. R. (2017, February). Punjabi Poetry Classification: The Test of 10 Machine Learning Algorithms. In Proceedings of the 9th International Conference on Machine Learning and Computing (pp. 1-5). ACM.

[10] Kaur, J., & Saini, J. R. (2017). PuPoCl: Development of Punjabi Poetry Classifier Using Linguistic Features and Weighting. INFOCOMP, 16(1-2), 1-7.

[11] Kaur, J., & Saini, J. R. (2018). Automatic classification of Punjabi poetries using poetic features. International Journal of Computational Intelligence Studies, 7(2), 124-137.

[12] Kaur, J., & Saini, J. R. (2016). Automatic Punjabi poetry classification using machine learning algorithms with reduced feature set. International Journal of Artificial Intelligence and Soft Computing, 5(4), 311-319.

[13] Kaur, J., & Saini, J. Designing Punjabi Poetry Classifiers Using Machine Learning and Different Textual Features.

[14] Chandrakar, O., & Saini, J. R. (2016, October). Development of Indian weighted diabetic risk score (IWDRS) using machine learning techniques for type-2 diabetes. In Proceedings of the 9th Annual ACM India Conference (pp. 125-128). ACM.

[15] Audichya M.A., Saini J.R., "Computational Linguistic Prosody Rule-based Unified Technique for Automatic Metadata Generation for Marathi Poetry", proceedings of ICAIT-2019, in press, IEEE, USA.

[16] Audichya M.A. and Saini J.R., 2020, "Computational Linguistic Prosody Rule-based Unified Technique for Automatic Metadata Generation for Hindi Poetry", 1st IEEE International Conference on Advances in Information Technology, Karnatka, India, in press with IEEE.

[17] Saini J.R. and Kaur J., 2020, "Kāvi: An Annotated Corpus of Punjabi Poetry with Emotion Detection Based on 'Navrasa'", Procedia Computer Science, in press with Elsevier.

[18] Bafna P.B., Saini J.R.,2019, "Identification of Significant Challenges in the Sports Domain using Clustering and Feature Selection Techniques", 9th International Conference on Emerging Trends in Engineering and Technology on Signal and Information Processing (ICETET-SIP-19, Nagpur, India, in press with IEEE.

[19] Bafna P.B., Saini J.R.,2019, "Hindi Multi-document Word Cloud based Summarization through Unsupervised Learning", ", 9th International Conference on Emerging Trends in Engineering and Technology on Signal and Information Processing (ICETET-SIP-19), Nagpur, India, in press with IEEE.

[20] Bafna P.B., Saini J.R., 2019, "Scaled Document Clustering and Word Cloud based Summarization on Hindi Corpus, 4th International Conference on Advanced Computing and Intelligent Engineering, Bhubaneshwar, India, in press with Springer.

[21] Bafna P.B., Saini J.R., 2020, On Readability Metrics of Goal Statements of Universities and Brand-promoting Lexicons for Industries, 4th International Conference of Data Management, Analytics and Innovation (ICDMAI 2020), Delhi,India.

[22] Bafna P.B., Saini J.R., 2020, Identification of Significant Challenges Faced by Tourism and Hospitality Industry Using Association rules", 4th International Conference of Data Management, Analytics and Innovation (ICDMAI 2020), Delhi,India.

[23] Bafna P.B., Saini J.R., 2020,"Marathi Text Analysis using Unsupervised Learning and Word Cloud", International Journal of Engineering and Advanced Technology,9(3),in press.

[24] Venugopal G., Saini J.R., Dhanya P., Novel Language Resources for Hindi: An Aesthetics Text Corpus and a Comprehensive Stop Lemma List, International Journal of Advanced Computer Science and Applications, vol. 11(1), Jan. 2020, in press.

[25] Bafna, P., Pramod, D., & Vaidya, A. (2016, March). Document clustering: TF-IDF approach. In 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) (pp. 61-66). IEEE.

[26] Harrag, F., & El-Qawasmah, E. (2009, August). Neural Network for Arabic text classification. In 2009 Second International Conference on the Applications of Digital Information and Web Technologies (pp. 778-783). IEEE.

[27] Yu, B., Xu, Z. B., & Li, C. H. (2008). Latent semantic analysis for text categorization using neural network. Knowledge-Based Systems, 21(8), 900-904.

[28] Zhang, D., & Lee, W. S. (2003, July). Question classification using support vector machines. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (pp. 26-32).

[29] https://aajtak.intoday.in/sahitya-kavita.html.