Bhaavana: A Novel and Comprehensive Hindi Poetry Classifier Based on Emotions

Dr. Kaushika Pal[1], Dr. JatinderKumar Saini[2]
[1]Assistant Professor, Sarvajanik College of Engineering & Technology, Surat, Gujarat, India
[2] Professor & Director, Symbiosis Institute of Computer Studies and Research, Pune, Maharastra, India

Abstract: Emotions are the essence of humanity and they lead to various sensations in human beings. In traditional Indian literature, these complex emotions are represented through the notion of 'Rasa' (meaning emotion). For the current research, five such 'Rasa' namely 'Hasya' (comic), 'Karuna' (compassion), 'Shanta' (calmness), 'Shringar' (romance) and 'Veera' (courage) have been used to design a classifier called 'Bhaavana' (emotion) for Hindi poetry. Technically, this is a Natural Language Processing (NLP) quinary (i.e. five-category) classification task and we make use of various sub-tasks including Pre-processing, Tokenization, Stemming, Bag-of-Words (BOW), Feature Extraction, and Part-Of-Speech (POS) tagging. Three types of linguistic features namely Lexical features (LEX), Syntactic features comprising Part-of-Speech (POS) (i.e., LEX+POS), and Emotion specific Features (ESF) have been deployed towards the aim of designing an automatic Hindi Poetry Classifier. A corpus of more than 800 poems with these 5 emotions and comprising more than 1,000,00 words have been processed to obtain a lexical feature set comprising more than 73,000 unique unigrams. Additionally, Highest Rank features (HRF) have been found and experimented with LEX, LEX+POS, and ESF. The various Machine Learning (ML) algorithms used are Gaussian Naïve Bayes (GNB), Multinomial Naïve Bayes (MNB), Neural Network (NN), and Support Vector Machine (SVM) and experimentation results with LEX, LEX+HRF, LEX+POS and LEX+POS+HRF, ESF+HRF for each ML algorithm are presented. These results are still further fortified by the use of Frequency Distribution (FD), Term Frequency (TF), and Term Frequency-Inverse Document Frequency (TF-IDF) during the experimentation. It is concluded that LEX+HRF is the best feature, FD is the best weighing method and MNB is the best algorithm. These are respectively followed by ESF+HRF and LEX+POS+HRF. The average of k-fold cross-validation results gives the best performance to be 71.09%. K-fold cross-validation experiments show that ESF+HRF is a more stable feature set giving stable results across various folds.

Keywords: Classification; Emotion; Hindi; Naïve Bayes (NB); Neural Network (NN); Part-Of-Speech (POS); Poetry; Support Vector Machine (SVM)

(Received June 27th, 2023/ Accepted January 06, 2024)

1 INTRODUCTION

Diverse languages are used in India and Hindi is not just one of the official languages but also one of the most spoken languages in India. The use of the Hindi language electronically to communicate and write blogs, articles, news reports, poetries and all forms of Hindi literature is increasing on the web. Hindi poetry comprises written text to express emotions which are represented through 'Rasa' (meaning emotions) in Indian literature. The objective of the current research work is to classify Hindi poetry documents into five emotional classes or five 'Rasa(s)' namely 'Hasya' (comic), 'Karuna' (compassion), 'Shanta' (calmness), 'Shringar' (romance) and 'Veera' (courage). This is motivated by the possibility of such a system's used to efficiently manage and easily retrieve Hindi poetry. Technically, this is an application of five-category, i.e. quinary document classification. Poem documents express emotions and have poetic elements like rhymes, metaphors, meter, sound, and imagination.

The automatic classification of Hindi Poetries based on emotions contributes to the field of Emotion Analysis as well as Text to Speech applications. Identification of emotions in written text aids to adjust frequency, tone, voice, and music to output speech in Text to Speech applications. This research work is using linguistic features, which are also called textual features, to classify the Poetries into 'Rasa(s)'. The morphology of Indian languages makes such processing challenging and exciting. The automatic classification of poems will support organizing and maintaining a large collection of poetry available on the world wide web. This will support the easy retrieval of poems and will be helpful to people who study literature. Labeled poetry supports indexing for easy poetry retrieval. The corpus created for this research is created using a web scraping module, which is automatized and can be used to extract more poetries with minimal change [14], and therefore supports information filtering, filtering of poetries from rest unnecessary elements on the web. The meaning and related emotions of ‘Rasa(s)’ are shown in detail in Table 1.

Table 1. ‘Rasa(s), and their meanings and associated emotions.

Sr. No.	Rasa(s)	Meaning & associated Emotions
1	‘Hasya’	Sarcasm, Joy, comic, Humor
2	‘Karuna’	Sympathy, Sadness, Compassion, Pity
3	‘Shanta’	Relaxation, peace, calmness
4	‘Shringar’	Romance, Love, Devotion, Beauty
5	‘Veera’	Confidence, Courage, Pride, Heroic

Different researchers have used different algorithms in an attempt to classify poetry in different languages but the formal classification of Hindi poetries based on emotions, specifically Rasa has been a rather unexplored area. This research paper is structured as follows. The introduction is followed by a review of related literature. This is followed by a discussion of the Methodology of the proposed approach, followed by the section comprising the presentation of the Results and Discussion. The paper ends with concluding remarks in the Conclusion section. We also present the future directions for the current research.

2 Literature Survey

Rakshit et al. [1] worked on the classification of Bangla poems of 4 classes namely ‘Pooja’, ‘Prem’, ‘Prakriti’, and ‘Swadesh’ with a dataset of 1341 Poems using SVM and achieved an accuracy of 56.80%. Alsharif et al. [2] worked on Arabic poems with 4 classes ‘Retha’, ‘Ghazal’, ‘Heja’, and ‘Fakhrtotal’ with a dataset of 1231 Poems using SVM, NB, Voting Feature Intervals, and Hyperpipes and achieved results in an accuracy of 79% with Hyperpipes. Noah et al. [3] worked on Malay Poems of 10 classes with a dataset of 1500 Poems using SVM with TF/IDF and achieved an accuracy of 58.44%. Kumar and Minz [4] worked on English poems of 8 categories with a dataset of 400 poems using SVM, NB, and KNN. Achieved results in an accuracy of 93.25% with SVM. Hamidi et al. [5] worked on Persian poems for 12 classes with 136 Poetries using SVM and achieved an accuracy of 91%. Anne et al. [6], worked on patent document classification and sub-classification using Machine Learning techniques, they used kNN, SVM, J48, and Random Forest and found SVM is giving better results with an accuracy of 69.2%.

Rennie et al. [7] claim that the Support Vector machine performs better in Multi-class Text Classification compared to Naïve Bayes. Chih-Chang and Lin [8] have discussed the ‘One vs All’ and ‘One vs One’ SVM for Multiclass classification. Gaur and Yadav [9] built a method to recognize scanned handwritten Hindi characters. After Binarizing the image, separating the characters, and removing horizontal bars they used SVM for recognizing the character. Puri and Singh [10] have used SVM to propose a classification model for Hindi printed and handwritten documentation. In another work, the same researchers, Puri and Singh [11] proposed a model for Hindi Text Classification, which uses SVM and accomplished experiments with only 4 documents of 2 classes. Kaur and Saini [12] have surveyed research done in Indic languages. In another work [13], they classified Punjabi poetries into 4 classes with maximum accuracy of 72.15% with SVM. Omar [16] has presented an analysis of the theme of the poetry of Dickinson by using the traditional concept of the Vector Space Document Model (VSDM). Pal and Patel [17] attempted to create a corpus of more than 1000 Hindi poetry documents belonging to different emotions. Kernot et al. [18] also worked in the area of poetry but with the objective of author identification for a given poem. Ternate et al. [19] used Hidden Markov Model (HMM) for the creation of poems. The poems generated by them are very small poems.

Poetry classification implemented in various languages majorly has used Support Vector Machine and Lexical features. Poem classification for various languages and Multiclass Text Classification in various Indian and Foreign Languages motivated us to carry the classification of Hindi Poetries based on complex ‘Rasa(s)’ as per Indian Literature.

Table 2 shows a focused summary of the work carried out for classification specifically of Poetries. A literature survey shows that no formal attempt has been made to explore the complex emotion theory for the classification of Hindi Poetry content.

Table 2. Summary of Related work

Sr. No.	Reference	Data Set[a]	Language	No. of Classes	Algorithm(s)	Accuracy[b]
1	Rakshit et al. [1]	1341	Bangla	4	SVM	56.80%
2	Alsharif et al. [2]	1231	Arabic	4	SVM, NB, HP	79% (HP)
3	Noah et al. [3]	1500	Persian	12	SVM	91%
4	Kumar and Minz [4]	400	Punjabi	4	HP, NN, NB, SVM, PART, C4.5, AB, BG, VFI, ZR	58.79% (SVM)
5	Hamidi et al. [5]	136	Punjabi	4	HP, KNN, NB, SVM	72.15% (SVM)
6	Kaur and Saini [12]	240	Hindi	4	SVM, DT, NN, NB	96% (SVM)
7	Kaur and Saini [13]	2034	English	3	SVM, LDA	84.80%
8	Bafna and Saini [20]	697	Spanish	4	DT	75.13%
9	Lou [21]	7214	Ottoman	10	SVM, NB	90% (SVM)
10	Barros [22]	185	Gujarati	9	Deep Learning	~87.62%
11	Can [23]	Collection of poems of ten poets	English	2	ZeroR, OneR	94.39%
12	Mehta et al. [24]	300+	Marathi	5	SVM	93.54%
13	Tanasescu et al. [25]	4986	English	2	Attention-based C-BiLSTM	88%
14	Deshmukh et al. [26]	341	Malaysian	9	SVM	89%
15	Ahmad et al. [27]	279	English	2	CNN	96.51%
16	Lou et al. [28]	7214	English (from diverse regions USA, Russia, China, UK, India, Greece)	13	KNN	Metaphor (Precision – 0.782, Recall – 0.822, F-Score – 0.781) Literal (Precision – 0.731, Recall – 0.726, F-Score – 0.711)
17	Peri-Polonijo [29]	12,830	English	2	CNN	96.51%
18	Kesarwani [30]	12,830 PoFo	English (from diverse regions USA, Russia, China, UK, India, Greece)	13	KNN	Metaphor (Precision – 0.782, Recall – 0.822, F-Score – 0.781) Literal (Precision – 0.731, Recall – 0.726, F-Score – 0.711)

[a]Data-set indicates specifically the number of 'poems', rather than any other type of data
[b]Best performing algorithm, for cases with multiple algorithms, is mentioned in parenthesis

3 Methodology

The classification model is built through various stages including

1. Data Pre-processing,
2. Part-Of-Speech (POS) tagger,
3. Feature Extraction,
4. Training,
5. and Testing, as shown in figure 1.

For the first time, grammar-based features are extracted from POS-tagged files as shown in figure 2. and are also used for the classification of poetry.

Total of 9 experiments with different feature sets was performed to find the best and most reliable classifier for Hindi poetries. 6 experiments are performed on unbalanced corpus, that is, the number of poetries differ in each category and 3 experiments are performed on balanced corpus.

1. **The Preprocessing stage** of the model comprises methods used to clean Poetry

Documents, Tokenize the Documents, Remove Stop Words and perform Stemming.

The Preprocessing module and POS Tagger module have been designed and implemented to perform bulk pre-processing and bulk POS tagging of the documents.

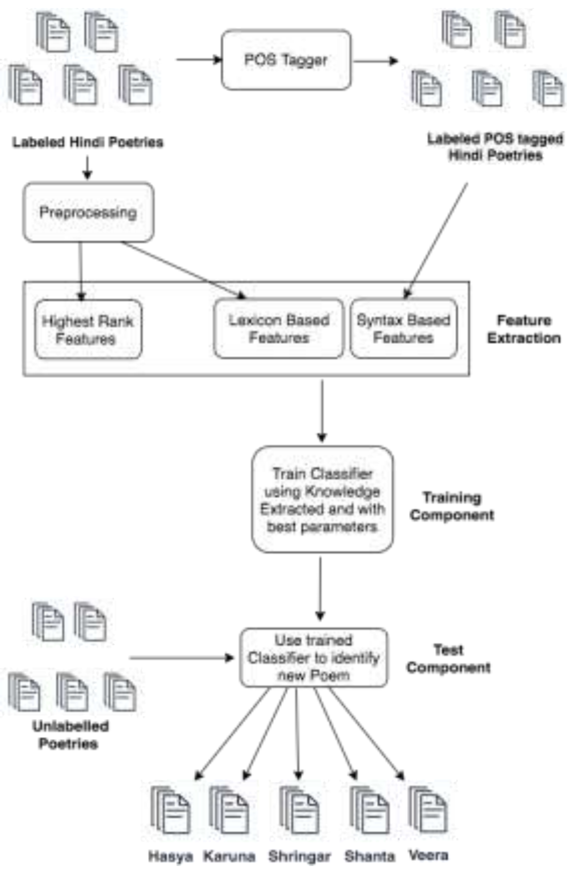


Figure 1. Architecture of Classification Model

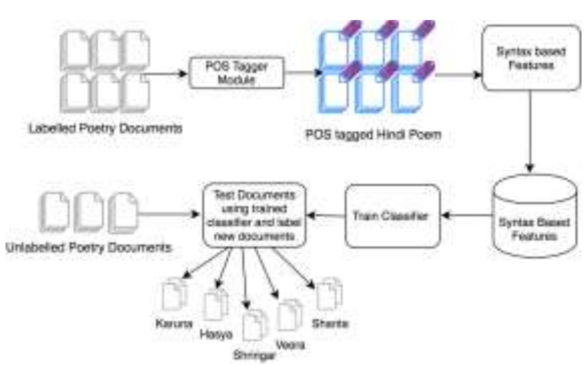


Figure 2. Architecture of classification model with syntax based feature extraction

The documents contain Hindi poems, one document represents one poem. For experimentation each rasa (category), each poetry, and each feature is converted in numeric form. Some features with corresponding numeric representation are shown in figure 3.

Feature	Numeric Representation	Feature	Numeric Representation	Feature	Numeric Representation	Feature	Numeric Representation
[illegible]	638	रात	1079	लहर	1049	[illegible]	1069
गमन	695	आग	148	पहर	770	अमर	107
[illegible]	819	[illegible]	806	[illegible]	506	[illegible]	503
[illegible]	1215	[illegible]	779	[illegible]	161	रस	997
उमड़	271	[illegible]	827	[illegible]	1177	[illegible]	525
[illegible]	804	पल	759	[illegible]	257	[illegible]	394
[illegible]	1033	[illegible]	215	[illegible]	439	दशा	656
[illegible]	274	[illegible]	1213	दम	647	[illegible]	404

Figure 3. Features with numeric representation

After numeric representation of each poetry, each category and each feature, Document Term Matrix (DTM) is created for each model to build an automatic classifier. The size of the DTM matrix is a number of poetries i.e., 830 by a number of features i.e., 73150 when using a bag of words model, which represents the presence of a feature in a particular poem. Figure 4 shows a document matrix of the occurrence of a feature with number 0 and 1, 1 represents the presence of a feature, and 0 represents the absence of the feature in the corresponding Poem document. The poetry and features are represented with numbers in figure 4.

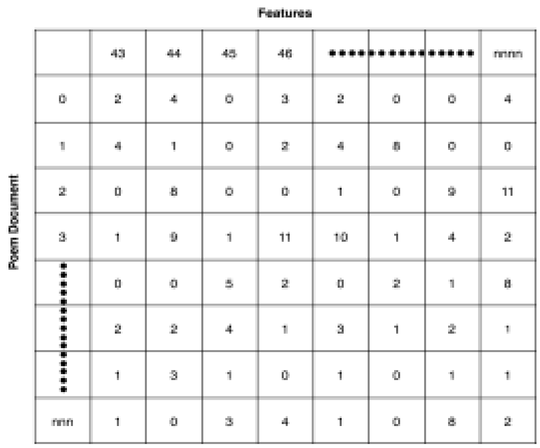
Features

Poem Document

P.Doc/ Feature	51	52	53	•••	nnnn
0	1	0	1	1	0
1	1	0	1	1	0
2	0	0	1	0	1
⋮	0	1	1	1	0
nnn	1	1	0	0	1

Figure 4. Document matrix with presence and absence of feature

The occurrence is replaced with the actual occurrence of the feature in the poem document and the same is shown in figure 5.



Features

	43	44	45	46	••••••••••••••			nnnn
0	2	4	0	3	2	0	0	4
1	4	1	0	2	4	8	0	0
2	0	8	0	0	1	0	9	11
3	1	9	1	11	10	1	4	2
•••	0	0	5	2	0	2	1	8
•••	2	2	4	1	3	1	2	1
•••	1	3	1	0	1	0	1	1
nnn	1	0	3	4	1	0	8	2

Poem Document

Figure 5. Document matrix with number of occurrence of feature

2. **The POS tagger** is used to tag each word of the poetry with its corresponding grammatical tags. It uses the structure of the poetry as well as the placement of words in the sentences to extract the syntactical features.

The Trigrams'n'Tags tagger [15] is trained using 'Hindi.pos' in order to tag each Hindi poetry document. The tags used by the POS-tagger are Noun, Pronoun, Adjective, Noun Location, Adverb, Intensifier, Conjunction, Postposition, Particle, Auxiliary Verb, Compound, Quantifier, Verb Finite Main, Negative, Punctuation, Compound Proper Noun, Noun in 'Kriya Mula', Question word, Verb Non-finite and Unknown.

3. **Feature extraction**

The next stage of the model is Feature Extraction and it extracts **three** types of **linguistic features**, namely 1. Lexical features (LEX), 2. Syntactical features (LEX+POS), and 3. Grammar bases features (ESF), and to boost the classification algorithm highest rank features are used.

The LEX feature set is obtained by using the Bag of Words (BOW) model for the pre-processed set of documents. The statistics for the LEX stage are presented in Table 3. We have called the words extracted during this stage of processing as tokens. The obtained token set contains words, which are unique, unigrams, non-stop-word, and stemmed. This is also the reason behind the number of words being reduced to 73,150 from an initial count of 1,055,59, as can be observed in Table 3. The time duration to achieve the token is presented in milliseconds (ms) in Table 3.

Table 3. Statistics of Lexical Feature Extraction

'Rasa(s)'	No. of Poems	Actual Tokens	Tokens Extracted	Duration (ms)
'Hasya'	105	10114	6766	202
'Karuna'	93	13781	9133	342
'Shanta'	272	31643	31643	1035
'Shringar'	216	22700	16621	622
'Veera'	144	27321	18668	561
Total	830	105559	73150	2762

The words, each with its POS tag, have been used as Syntactical features, that is **LEX+POS** feature set. The statistics of the time consumed by the tagging process are shown in Table 4 . The time duration is shown in minutes and seconds (mm: ss) in Table 6.

The pseudocode for POS-tagging of the poetry of a given directory is shown in Table 5. LEX+POS is obtained by consideration of each LEX feature additionally along with its POS-tag, as obtained through the implementation of the POS-tagger.

Table 4. Statistics of POS Tagging

'Rasa(s)'	Poems	Duration
'Hasya'	105	01:59
'Karuna'	93	02:50
'Shanta'	272	05:50
'Shringar'	216	03:36
'Veera'	144	07:29
Total	830	21:44

Table 5. Pseudocode for Bulk Tagging of Poetries

```
SET NoofDocs as 0
SET TaggedText as NULL LIST
FOR each file in Directory:
    IF file ends with (.txt)
    Initialize Text with content of the file
    FOR each token in Text
            Tag each token of Text with POS tag
            STORE and Append the result in TaggedText
    ENDFOR
    ENDIF
    WRITE TaggedText in file
    SET TaggedText to NULL LIST
    Increment NoOfDocs by 1
ENDFOR
```

For Emotion specific features (ESF) feature set Grammar-based features are extracted from POS-tagged files. Adjectives, Adverbs, Nouns, and Intensifiers are extracted from POS-tagged files, and other Unknown elements tagged as 'Unk' are also extracted, cleaned, and used for creating emotion-specific features (ESF). The pseudocode to extract grammar-based features is shown in Table 6 and the statistics for the same are shown in Table 7.

Table 6. Pseudocode to find Grammer Rank Features(ESF) from Poetries.

```
SET NoofPoems as 0
```

2. SET FeatureList as NULL LIST
3. Set grammar rules with Adjectives, Adverbs, Nouns, Intensifiers and Unk
4. Create chunk parser with grammar rules
5. FOR each file in Directory(Rasa(s)Name):
6. Open and read POS tagged Poem
7. Populate FeatureList by extracting features using chuck parser.
8. Write FeatureList in file
9. Move to next file
10. ENDFOR

Table 7. Statistics of Grammar based Features extracted from POS Tagged Poems.

‘Rasa(s)’	No. of Poems	Actual Tokens	ESF Extracted	Duration (mm: ss)
‘Hasya’	105	10114	4927	01:50
‘Karuna’	93	13781	7046	02:38
‘Shanta’	272	31643	17764	05:20
‘Shringar’	216	22700	14275	03:20
‘Veera’	144	27321	15247	07:01
Total	830	105559	59259	20:15

To boost the classification algorithm, the ‘Rasa(s)’-wise poetry documents are used to find the Highest Rank Features (HRF) which are found by using the top 20 occurring features in each ‘Rasa’ across all poems belonging to that ‘Rasa’. The pseudocode to find Highest Rank Features is given in Table 8. This HRF is then added with LEX and LEX+POS to further enhance the classification performance.

Table 8. Pseudocode to find Highest Rank Features(HRF) from Poetries.

1. SET HRF as NULL LIST
2. SET NoofPoems as 0
3. SET FeatureList as NULL LIST
4. SET R as 20
5. FOR each file in Directory(Rasa(s)Name):
6. Open and read Poem
7. Populate FeatureList with features and its occurrence in poem.
8. Sort FeatureList in descending order of occurrence of feature.
9. Populate HRF by extracting top R features from FeatureList
10. Open and append HRF in file
11. Move to next file
12. ENDFOR

The model is finally built with five different types of feature sets namely, LEX, LEX+HRF, LEX+POS, LEX+POS+HRF, and ESF+HRF.

The machine learning algorithms finalized to use for implementing the Hindi Poetry classifier based on the notion of ‘Rasa’s’ and these five feature sets are Gaussian Naïve Bayes (GNB), Multinomial Naïve Bayes (MNB), Neural Network (NN) and Support Vector Machine (SVM) are selected after testing 14 different algorithms combinations on the full corpus.

The models are further fortified by experimentation with Frequency Distribution (FD), Term Frequency (TF), and Term Frequency-Inverse Document Frequency (TF-IDF).

4. The **Training stage** is used for training the model with the extracted features by using one of the features at a time (e.g. first LEX, then LEX+HRF, etc.)

5. and the **Testing stage** is used to classify the model-unseen poem document into one of the five ‘Rasa’ categories.

Keeping in view the orientation of this research work towards ‘emotions’ and the use of Hindi language poetry, we have called the proposed model with the name ‘Bhaavana’ (‘□□□□□’, meaning ‘emotions’ in Hindi).

The performance of the model is evaluated with Accuracy, Precision, Recall, Precision-Recall Curve (PRC), and F1-score. The area of PRC is calculated by plotting Precision and Recall for a two-dimensional plane on Y-axis and X-axis, respectively. The confusion matrix is used to depict the performance of the classifier. All these results are presented in the following section.

4 Experiments, Results and Discussion

For the very first time in the research community, an attempt has been made to design and implement a multi-class quinary classifier for the Hindi Poetry documents with grammar-based features (ESF), syntax-based features (LEX+POS), and lexical features (LEX) supported by Top rank features. The entire implementation as well as the execution of all the experiments for various stages and sub-stages described in the Methodology section was done using Python 3.6 on a macOS High Sierra version 10.13.1. The used CPU is a 1.8 GHz Intel Core i5 with 8 GB 1600 MHz DDR3 memory. For better performance, instead of using the traditional Hard Disk Drive (HDD) storage device, the files were stored on a faster Solid-State Drive (SSD).

The corpus for this research work has been created by web scraping the labeled poetries from the poetry collection available for free at [14]. A total of 830 poems are extracted with ‘Hasya’ category having 144 poems and ‘Karuna’ category having 93 poems. Similarly, the poems in other categories namely ‘Shanta’, ‘Shringar’, and ‘Veera’ are 272, 216, and 105 respectively.

Experiments

A total of 9 experiments performed to find the best-automated classifier are shown in table 9 including 5 different feature sets. The feature sets used were Complete Poem Lexical (CPL), LEX, LEX + POS, LEX

+ HRF, LEX + POS + HRF, ESF + HRF with 830 Poetries and feature weighing used were FD, TF and ITF and LEX + HRF, LEX + POS +HRF, ESF + HRF on balanced corpus of 501 poetries using FD feature weighing. The results are shown in table 9

For the initial experimentation, the corpus was divided randomly to prepare two sets in a proportion of 80% and 20% respectively for the training and testing data sets..

Table 9. Maximum Accuracy achieved on full corpus with different feature sets.

Experiment No.	Feature Set	Corpus Size	Feature Weighing	No. of Algorithm	Maximum Acuracy
1	CPL	830	FD, TF, ITF	14	51.49% with KNN
2	LEX	830	FD, TF, ITF	4	57.83% with SVM & SGDC
3	LEX + HRF	830	FD, TF, ITF	4	69.46% with MNB
4	LEX + POS	830	FD, TF, ITF	4	56.62% with SGDC
5	LEX + POS + HRF	830	FD, TF, ITF	4	64.67% with MNB
6	ESF + HRF	830	FD, TF, ITF	4	67.86% with MNB
7	LEX + HRF	501	FD	2	77.45% with MNB
8	LEX + POS + HRF	501	FD	2	71.56% with GNB
9	ESF + HRF	501	FD	2	71.56% with MNB

The comparison of results on 5 different feature sets with 4 different Machine Learning algorithms is shown in table 10.

The results of MNB – multinomial Naïve Bayes are best across all feature sets.

Table 10. Comparison of results of accuracy (in %) for Full Corpus with different feature sets.

Algorithm	LEX	LEX+HRF	LEX+POS	LEX+POS+HRF	ESF+HRF
GNB	40.36	57.48	40.36	57.48	53.57
MNB	56.02	**69.46**	57.22	**64.67**	**67.86**
NN	51.20	47.90	42.16	50.89	56.54
SVM	47.59	50.29	51.80	48.50	53.57

The confusion matrix and precision-recall curve for the LEX+HRF feature set are shown in Figures 6 and 7 respectively.

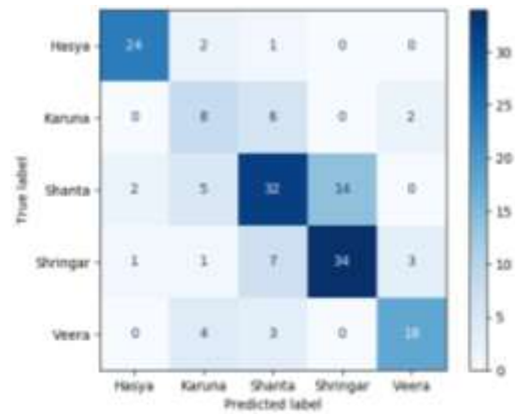


Figure. 6. Confusion Matrix for LEX+HRF with MNB

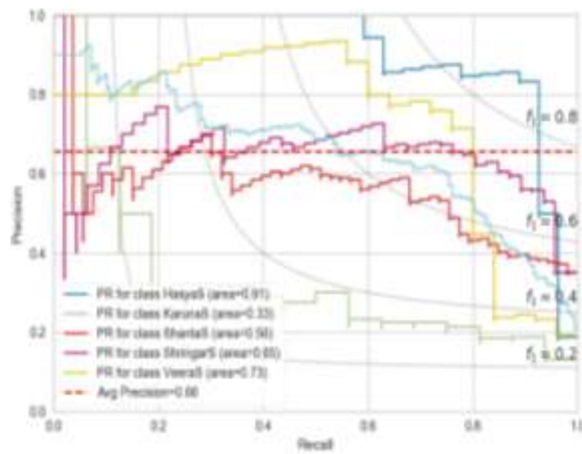


Figure 7. Precision Recall Curve for LEX+HRF with MNB

Results and Discussion

The performance of feature sets LEX+HRF, LEX+POS+HRF, and ESF+HRF are better in comparison to LEX and LEX+POS, which is clearly visible in table 11. F1-score, PRC area, and Accuracy across all 5 Rasa on complete corpus are shown in table 11.

Looking at the results of MNB and GNB, additional experiments were conducted on 501 poetries comprising 105 'Hasya', 93 'Karuna', 101 'Shringar', 101 'Shanta', and 101 'Veera' 'Rasa' by nearly balancing the number of poetry in each category and we call these experiments to be with 'balanced-corpus'.

Results of Balanced corpus

The experiment was conducted for MNB on LEX+HRF, LEX+POS+HRF, and ESF+HRF feature sets further fortified by Frequency Distribution (FD), Term Frequency (TF), and Term Frequency-Inverse Document Frequency (TF-IDF). The same experiment was repeated with all the same parameters but for GNB.

The results of both these experiments are presented in Table 12. Table 13 and Table 14 respectively present the results of the performance evaluation of GNB and MNB on Balanced Corpus with the LEX+HRF feature set.

Table 12 is showing comparison of accuracy with LEX+HRF, LEX+POS+HRF and ESF+HRF feature set with FD, TF and TF-IDF weighing measures using GNB and MNB algorithms. Table 13 shows Rasa wise performance using GNB algorithm with LEX+HRF feature set and we use F1-Score, Area of PRC and accuracy to see rasa wise performance of the classifier built with GNB algorithm. Similar rasa wise results with MNB are shown in Table 14. From Table 13 and Table 14, it can be seen that with the LEX+HRF feature set, overall TF-IDF performs better for GNB while overall FD performs better for MNB.

These Rasa wise performance was evaluated for GNB and MNB algorithms on the LEX+POS+HRF feature set and class wise results are shown in table 15 and table 16 respectively with FD, TF and TF-IDF weighing methods.

From Table 15 and Table 16, it can be seen that with the LEX+POS+HRF features set, overall both TF-IDF and FD perform better for GNB while overall FD performs better for MNB.

The results from all four tables from Table 13 to Table 16 could be generalized to say that irrespective of the feature set, TF-IDF always performs well for GNB while FD always performs well for MNB. The confusion matrix and PRC of the model built with MNB on balanced Corpus using LEX+HRF are shown in Fig. 8 and Fig. 9 respectively.

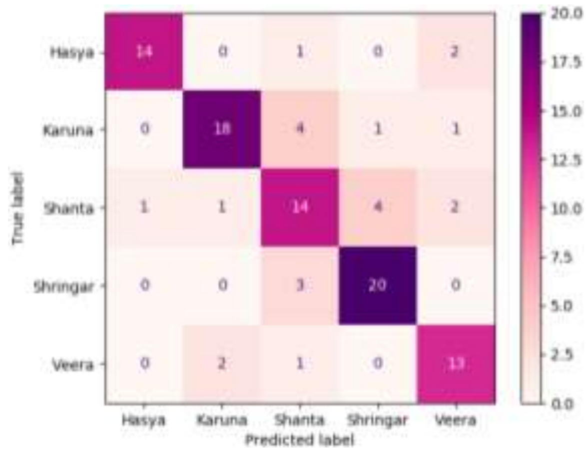


Figure 8. Confusion Matrix for LEX+HRF with MNB on Balanced Corpus

Table 11. Performance of MNB on full corpus

'Rasa(s)'	LEX+POS+HRF			LEX+HRF			ESF+HRF		
	F1-Score	PRC	Accuracy	F1-Score	PRC	Accuracy	F1-Score	PRC	Accuracy
'Hasya'	0.86	0.94	77.78%	0.89	0.91	88.89%	0.86	0.88	81.82%
'Karuna'	0.37	0.34	43.75%	0.44	0.33	50.00%	0.50	0.44	44.44%
'Shanta'	0.53	0.54	64.15%	0.63	0.56	60.38%	0.68	0.61	79.25%
'Shringar'	0.67	0.61	63.04%	0.72	0.65	73.91%	0.65	0.71	60.87%
'Veera'	0.71	0.73	68.00%	0.72	0.73	72.00%	0.68	0.71	69.23%
Total	0.66	0.63	**64.67%**	**0.69**	**0.66**	**69.03%**	0.68	0.68	**67.86%**

Table 12. Performance of Accuracy for GNB and MNB on Balanced Corpus

Algorithm	LEX+HRF			LEX+POS+HRF			ESF+HRF		
	FD	TF	TF-IDF	FD	TF	TF-IDF	FD	TF	TF-IDF
GNB	66.66%	63.72%	67.65%	71.56%	65.68%	71.56%	67.64%	63.72%	63.72%
MNB	**77.45%**	50.98%	57.84%	**70.58%**	45.09%	57.84%	**71.56%**	45.09%	53.92%

Table 13. Performance of GNB on balanced corpus with LEX+HRF feature set

'Rasa(s)'	F1-Score	Area of PRC	Accuracy in (%)

	FD	TF	TF-IDF	FD	TF	TF-IDF	FD	TF	TF-IDF
'Hasya'	0.80	0.80	0.87	0.75	0.75	0.75	70.59	70.59	76.47
'Karuna'	0.63	0.63	0.63	0.46	0.38	0.43	54.17	54.17	54.17
'Shanta'	0.53	0.49	0.53	0.37	0.31	0.34	59.09	54.55	59.09
'Shringar'	0.82	0.74	0.85	0.63	0.60	0.60	86.96	73.91	86.96
'Veera'	0.57	0.58	0.54	0.52	0.52	0.48	62.50	68.75	62.50
Total	0.67	0.64	**0.68**	0.52	0.47	0.49	66.66	63.72	**67.64**

Table 14. Performance of MNB on balanced corpus with LEX+HRF feature set

'Rasa(s)'	F1-Score			Area of PRC			Accuracy in (%)		
	FD	TF	TF-IDF	FD	TF	TF-IDF	FD	TF	TF-IDF
'Hasya'	0.87	0.67	0.67	0.88	0.75	0.75	**82.35**	70.59	70.59
'Karuna'	0.80	0.41	0.56	0.83	0.39	0.50	**75.00**	29.17	41.67
'Shanta'	0.62	0.29	0.34	0.53	0.26	0.31	**63.64**	27.27	31.82
'Shringar'	0.83	0.57	0.71	0.92	0.73	0.76	**86.96**	52.17	69.57
'Veera'	0.76	0.59	0.61	0.81	0.43	0.54	81.25	**93.75**	87.50
Total	**0.77**	0.51	0.58	**0.80**	0.49	0.55	**77.45**	50.98	57.84

Table 15. Performance of GNB on balanced corpus with LEX+POS+ HRF feature set

'Rasa(s)'	F1-Score			Area of PRC			Accuracy in (%)		
	FD	TF	TF-IDF	FD	TF	TF-IDF	FD	TF	TF-IDF
'Hasya'	0.87	0.87	0.87	0.75	0.75	0.75	76.47	76.47	76.47
'Karuna'	0.67	0.56	0.65	0.34	0.32	0.32	58.33	45.83	54.17
'Shanta'	0.62	0.57	0.62	0.36	0.31	0.32	68.18	63.64	68.18
'Shringar'	0.80	0.57	0.83	0.63	0.59	0.59	86.96	78.26	86.96
'Veera'	0.65	0.56	0.63	0.57	0.52	0.52	68.75	68.75	75.00
Total	**0.72**	0.66	**0.72**	**0.49**	0.46	0.46	**71.56**	65.68	**71.56**

Table 16. Performance of MNB on Balanced Corpus with LEX+POS+HRF feature set

'Rasa(s)'	F1-Score			Area of PRC			Accuracy in (%)		
	FD	TF	TF-IDF	FD	TF	TF-IDF	FD	TF	TF-IDF
'Hasya'	0.90	0.64	0.74	0.94	0.66	0.79	82.35	94.12	82.35
'Karuna'	0.69	0.15	0.56	0.84	0.25	0.41	70.83	08.33	45.83
'Shanta'	0.49	0.22	0.38	0.59	0.39	0.35	40.91	13.64	36.36
'Shringar'	0.73	0.58	0.64	0.82	0.55	0.66	78.26	47.83	60.87
'Veera'	0.74	0.44	0.59	0.80	0.35	0.48	87.50	87.50	75.00
Total	**0.71**	0.45	0.58	**0.78**	0.40	0.51	**70.58**	45.09	57.84

Table 17. Performance of GNB and MNB on Balanced Corpus with ESF+HRF feature set with FD

'Rasa(s)'	GNB			MNB		
	F1-Score	Area of PRC	Accuracy (in %)	F1-Score	Area of PRC	Accuracy (in %)
'Hasya'	0.91	0.84	1.00	0.97	0.98	1.00
'Karuna'	0.58	0.28	0.45	0.65	0.69	0.58
'Shanta'	0.55	0.33	0.60	0.63	0.56	0.65
'Shringar'	0.82	0.64	0.88	0.75	0.88	0.69
'Veera'	0.45	0.24	0.43	0.62	0.62	0.75
Total	**0.67**	0.43	67.64	**0.72**	0.75	71.56

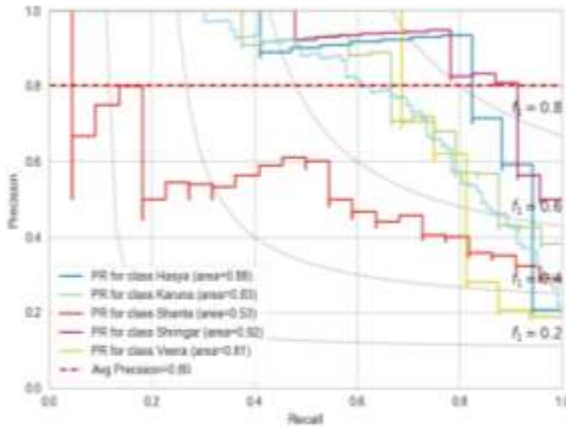


Figure 9. Precision Recall Curve for LEX+HRF with MNB on Balanced Corpus

Similarly, the confusion matrix and PRC for LEX+POS+HRF with GNB on Balanced Corpus are shown in Fig. 10 and Fig. 11 respectively.

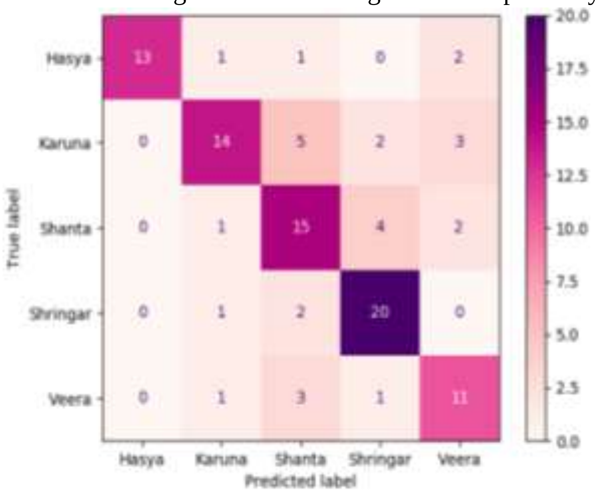


Figure 10. Confusion Matrix for LEX+POS+HRF with GNB on Balanced Corpus.

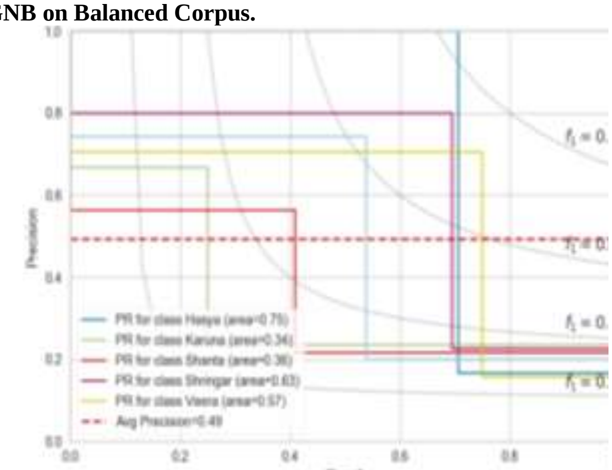


Figure 11. Precision Recall Curve for LEX+POS+HRF with GNB on Balanced Corpus

The confusion matrix for LEX+POS+HRF with MNB on Balanced Corpus as well as PRC for LEX+POS+HRF with MNB on Balanced Corpus is presented in Fig. 12 and Fig. 13, respectively.

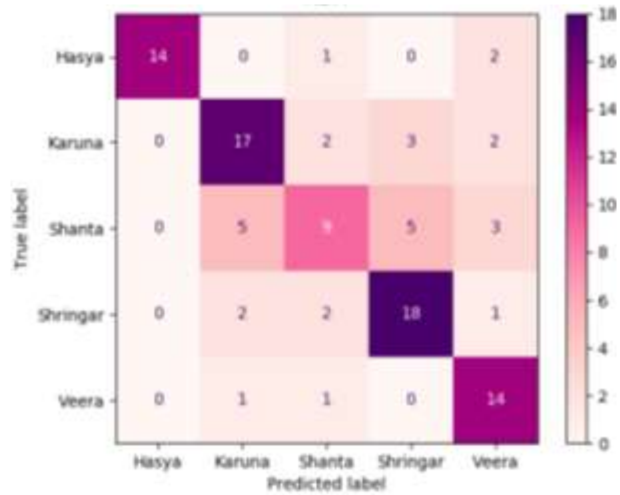


Figure 12. Confusion Matrix for LEX+POS+HRF with MNB on Balanced Corpus

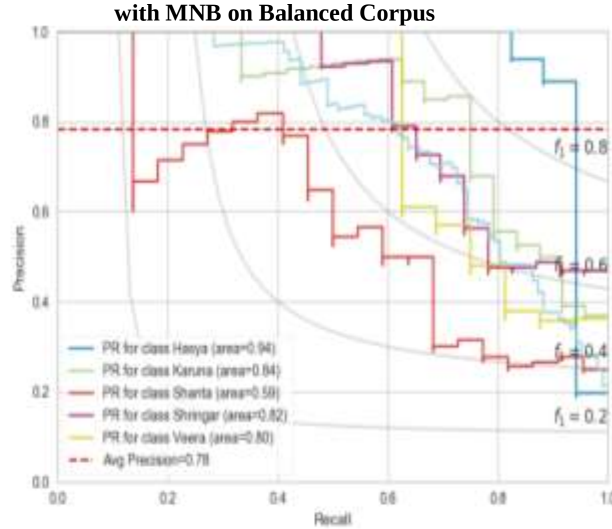


Figure 13. Precision Recall Curve for LEX+POS+HRF with MNB on Balanced Corpus

Finally, the rasa-wise performance of the ESF+HRF feature set for both GNB and MNB with FD are shown in table 17. The confusion matrix and PRC area with ESF+HRF using GNB are shown in Fig. 14 and Fig. 15 and the Confusion matrix and PRC area with ESF+HRF using MNB are shown in Fig. 16 and Fig. 17.

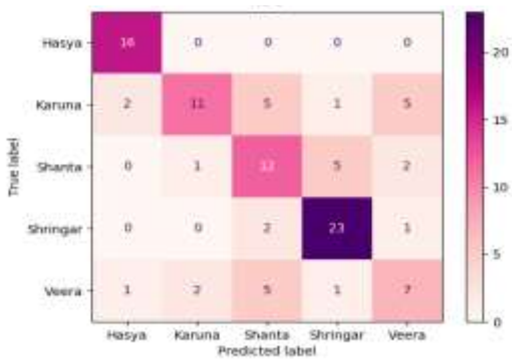


Figure. 14 Confusion Matrix for ESF+HRF with GNB on Balanced Corpus

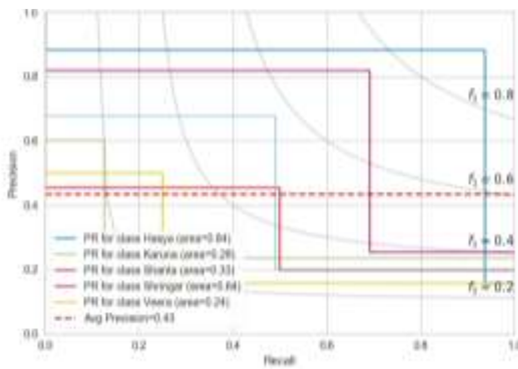


Figure 15. Precision Recall Curve for ESF +HRF with GNB on Balanced Corpus

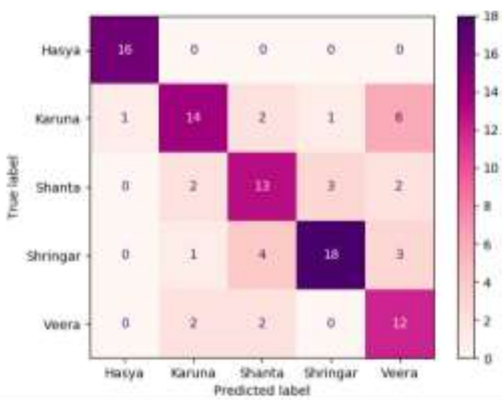


Figure 16. Confusion Matrix for ESF+HRF with MNB on Balanced Corpus

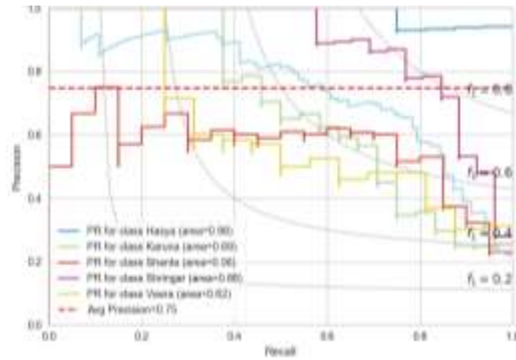


Figure 17. Precision Recall Curve for ESF +HRF with MNB on Balanced Corpus

The results in terms of maximum accuracy are achieved to be 66.66 and 77.45 using the LEX +HRF feature set with GNB and MNB algorithms respectively. We achieved 71.56% and 70.58% with GNB and MNB using the LEX+POS+HRF feature set. Using ESF+HRF with GNB and MNB the accuracy is 67.64% and 71.56%. All good results were achieved with FD. With the TF-IDF weighing method with GNB using LEX+POS+HRF the accuracy achieved is 71.56%.

The results showed that the best performing algorithm is MNB whereas the best performing feature is LEX+HRF followed by LEX+POS+HRF and ESF+HRF. It is noteworthy that two top-performing feature sets LEX+HRF and ESF+HRF features correspond to the MNB algorithm and GNB performs better for LEX+POS+HRF.

Also, since the result of LEX+HRF with an accuracy of 77.45%, is better compared to other feature combinations, more experimentation with resampling methods was used to validate the results. k-fold cross validation was applied on a balanced corpus and the number of folds was taken in the range of 4 to 20. The results of the accuracy of the experiments are shown in Table 18. The results for each value of k using the LEX+HRF feature with GNB and MNB is shown in the table.. From Table 18, it can be seen that the average accuracy for MNB (71.09%) is better than the average accuracy for GNB (65.96%). Further, the highest accuracy of 73.24% is also found for MNB. The box plot of the results across all folds for LEX+HRF is pictorially presented in Figure. 18.

Table 18. Average accuracy (in%) with k-fold cross validation on balanced corpus with LEX+HRF feature set

No. of folds (k)	GNB	MNB
4	63.61	68.06
8	66.01	70.29
10	67.23	71.01
12	66.05	71.29

16	66.50	**73.24**
18	66.04	71.75
20	66.26	71.99
Average	65.96%	71.09

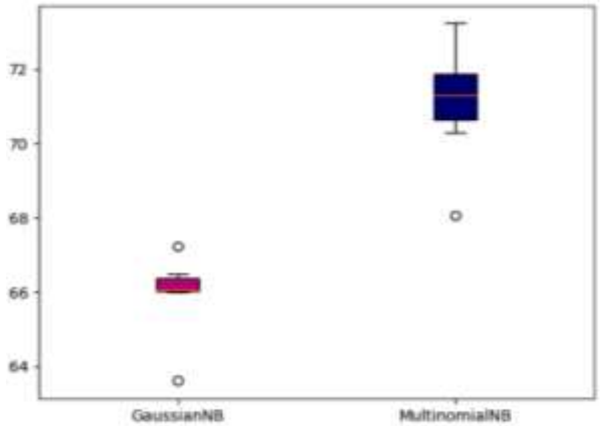


Figure 18: Range of accuracy using K-fold cross validation with LEX+HRF

The same experiment was performed for LEX+POS+HRF and ESF+HRF feature sets and the result is shown in Tables 19 and 20 respectively. The average accuracy with LEX+POS+HRF for GNB (68.89%) is better than the average accuracy for MNB (66.03%). The box plot of the results across all folds for LEX+POS+HRF is pictorially presented in Figure 19.

Table 19. Average accuracy (in%) with k-fold cross validation on balanced corpus with LEX+POS+HRF feature set

No. of folds (k)	**GNB**	**MNB**
4	67.07	63.86
8	68.50	64.35
10	69.00	66.57
12	69.28	66.59
16	69.48	66.78
18	69.45	67.32
20	69.48	66.79
Average	68.89	66.03

Table 20. Average accuracy (in%) with k-fold cross validation on Balanced corpus with ESF+HRF feature set

No. of folds (k)	**GNB**	**MNB**
4	63.09	71.45
8	63.08	71.20
10	63.09	71.20
12	63.81	71.97
16	62.34	72.15
18	64.13	72.43
20	64.79	71.90
Average	63.48%	71.75

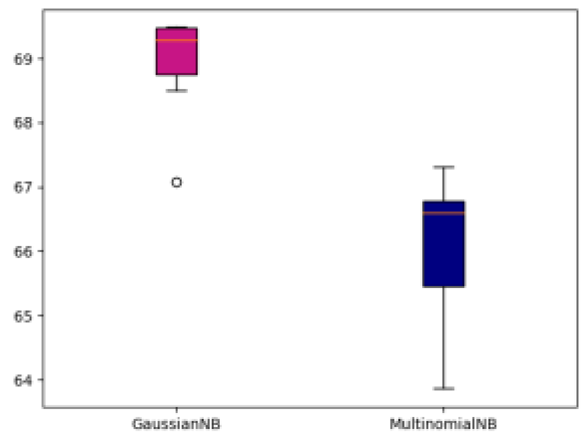


Figure 19: Range of accuracy using K-fold cross validation with LEX+POS+HRF

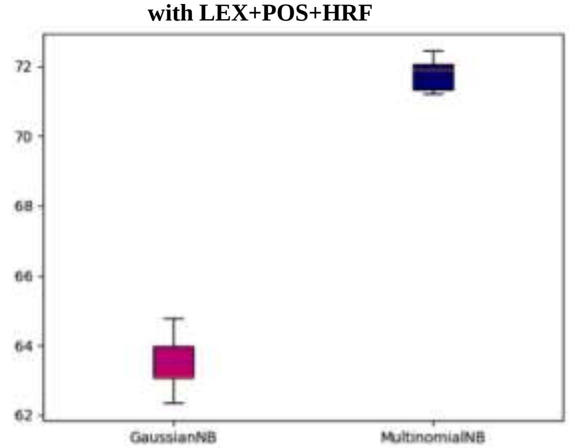


Figure 20: Range of accuracy using K-fold cross validation with ECF+HRF

The average accuracy with ESF+HRF for MNB (71.75%) is better than the average accuracy for GNB (63.48%). The box plot of the results across all folds for ESF+HRF is pictorially presented in Figure 20.

From experimentation, it is clear that with k-fold validation the maximum stable and consistent accuracy is found with the ESF+HRF feature set using MNB with only a difference of 1.23% across folds, while the LEX+HRF feature set using MNB differs by 5.18%. The results clearly state that for stable results the Emotion Specific feature with the highest rank (ESF+HRF) feature set is reliable.

5 Conclusion and Future Work

The proposed poetry classifier has experimented with 830 poems. The experimental results with full-corpus showed that LEX+HRF, LEX+POS+HRF, and ESF+HRF outperform other feature sets. Also, the efficiency of MNB and GNB was found to be better than other machine learning algorithms. It is concluded that LEX, LEX+POS, and ESF, all when fortified with HRF, yield better results. Based on the experimental results, it is also concluded that TF-IDF and FD perform well with GNB and MNB, respectively. More experimentation was done with all 5 categories having almost the same number of poems. We call it a balanced corpus and found that the obtained results were better compared to the unbalanced full corpus. The best performing feature set for the balanced-corpus was LEX+HRF. It is, hence, recommended that such experiments should be performed always with balanced datasets. The top three performing algorithms are concluded, in order, to be MNB, GNB, and SVM while FD and TF-IDF have been concluded to be the best weighing methods. Overall, we conclude that MNB is the best algorithm for the quinary classification of Hindi poetry into five emotion-based categories. 'Shringar' and 'Hasya', in order of decreasing rank, are the top best categories for accuracy of classification. The k-fold cross-validation results showed that the average best accuracy performance is 71.09% while the instance-based maximum best accuracy performance is 73.24%. It was also noticed that for more stable results considering any instance ESF+HRF gave robust results ranging from 71.20% to 72.43%.

This is the first of its kind work where various complex combinations of different machine learning algorithms, feature sets, and weighing methods have been experimented with for Hindi poetry classification which has been a rather unexplored area. The presented results will be definitely helpful to contemporary and future researchers not just in the field of Hindi poetry classification but for the poetry classification of various other languages as well. These results are reported based on the 'Rasa' or emotion-based categorization and the results could differ if there is a change in the size of the dataset or the number of emotions deployed for classification. In the future, we plan to perform more experiments with a greater number of emotions, more machine learning algorithms as well as different weighing factors. Such classification results could be used in Text to speech applications to fine tune the voice, rhythm and tone of the speech.

References

[1] Rakhsit G., Ghosh A., Bhattacharyya P., Haffari G., "Automated Analysis of Bangla Poetry for Classification and Poet Identification", in proceedings of 12th International Conference on Natural Language Processing, Trivandrum, India, 2015, pp. 247–253. Online: https://www.cse.iitb.ac.in/~pb/papers/icon15-poet-identification.pdf

[2] Alsharif O., Alshamaa D., Ghneim N., "Emotion Classification in Arabic Poetry using Machine Learning", International. Journal of Computer Application, vol. 65(16), 2013, pp. 10-15. doi: 10.5120/11006-6300

[3] Noah S.A., Jamal N., Mohd M., "Poetry classification using support vector machines", Journal of Computer Science, vol. 8(6), 2012, pp. 1441–1446. doi: 10.3844/jcssp.2012.1441.1446

[4] Kumar V., Minz S., "Poem Classification Using Machine Learning Approach", Advances in Intelligent Systems and Computing, vol. 236, 2012, pp. 675-682. doi: 10.1007/978-81-322-1602-5_72

[5] Hamidi S., Razzazi F., Ghaemmaghami M.P., "Automatic Meter Classification in Persian Poetries using Support Vector Machines", in proceedings of IEEE International Symposium on Signal Processing and Information Technology (ISSPIT-2009), 2009, pp. 563-567. doi: 10.1109/ISSPIT.2009.5407514

[6] Anne C., Mishra A., Hoque M.T., Tu S., "Multiclass Patent Document Classification", Artificial Intelligence Research, vol. 7(1), 2017, pp. 1-14. doi: 10.5430/air.v7n1p1

[7] Rennie J.D.M., Rifkin R., "Improving Multiclass Text Classification with the Support Vector Machine", in AI Memos of Massachuseets Institute of Technology, 2001. Online: https://dspace.mit.edu/handle/1721.1/7241

[8] Chang C.C., Lin C.J., "LIBSVM: A Library for Support Vector Machines", ACM Transactions on Intelligent Systems and Technology, vol. 2(3), 2011, pp. 1-27. doi: 10.1145/1961189.1961199

[9] Gaur A., Yadav S., "Handwritten Hindi Character Recognition using K- Means Clustering and SVM", in proceedings of 4th International Symposium on Emerging Trends and Technology in Libraries and Information Services, 2015. doi:10.1109/ettlis.2015.7048173

[10] Puri S., Singh S.P., "Hindi Text Document Classification System Using SVM and Fuzzy: A Survey", International Journal of Rough Sets and Data Analysis, vol. 5(4), pp. 1-31, 2018. doi: 10.4018/ijrsda.2018100101

[11] Puri S., Singh S.P., "An Efficient Hindi Text Classification Model Using SVM", Computing and Network Sustainability, vol. 75, 2019. doi: 10.1007/978-981-13-7150-9_24

[12] Kaur J., Saini J.R., "Punjabi Poetry Classification: The Test of 10 Machine Learning Algorithms", in proceedings of ACM International Conference on Machine Learning and Computing (ICMLC-2017), Singapore, 2017, pp. 1-5. doi:10.1145/3055635.3056589

[13] Kaur J., Saini J.R., "PuPoCl: Development of Punjabi Poetry Classifier Using Linguistic Features and Weighting", INFOCOMP Journal of Computer Science, Vol. 16(1-2), 2017, pp. 1-7. Online: http://infocomp.dcc.ufla.br/index.php/infocomp/article/view/546

[14] Hindi Poetry Collection. Online: https://maatribhasha.com/

[15] Trigrams'n'Tags tagger. Online: https://www.nltk.org/_modules/nltk/tag/tnt.html

[16] Omar A., "On the Digital Applications in the Thematic Literature Studies of Emily Dickinson's Poetry", International Journal of Advanced Computer Science and Applications, vol. 11(6), 2020, pp. 361-365. doi:10.14569/ijacsa.2020.0110647

[17] Pal K., Patel B.V., "Automatic Categorized Corpus Creation of Hindi Poetries Based on 'Rasa(s)' for Linguistics Research", Smart Innovation, Systems and Technologies, vol. 235, 2021, pp. 549-556. doi:10.1007/978-981-16-2877-1_50

[18] Kernot D., Bossomaier T., Bradbury R., "Stylometric Techniques for Multiple Author Clustering", International Journal of Advanced Computer Science and Applications, vol. 8(3), 2017, pp. 1-8. doi:10.14569/ijacsa.2017.080301

[19] Tarnate K.J.M., Garcia M.M., Sotelo-Bator P., "Short Poem Generation (SPG): A Performance Evaluation of Hidden Markov Model based on Readability Index and Turing Test", International Journal of Advanced Computer Science and Applications, vol. 11(2), 2020, pp. 294-297. doi:10.14569/ijacsa.2020.0110238

[20] Bafna P.B., Saini J.R., "On Exhaustive Evaluation of Eager Machine Learning Algorithms for Classification of Hindi Verses", International Journal of Advanced Computer Science and Applications, vol. 11(2), 2020, pp. 181-185. doi:10.14569/ijacsa.2020.0110224

[21] Lou A., Inkpen D., Tan C., "Multicategory Subject-Based Classification of Poetry", in proceedings of the 28th International Florida Artificial Intelligence Research Society Conference, 2015, pp. 187-192. Online: https://www.site.uottawa.ca/~diana/publications/flairs_2015_paper.pdf

[22] Barros L., Rodriguez P., Ortigosa A., "Automatic Classification of Literature Pieces by Emotion Detection: A Study on Quevedo's Poetry", in proceedings of Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII), 2013, pp. 141-146. doi:10.1109/ACII.2013.30

[23] Can E.F., Can F., Duygulu P., Kalpakli M., "Automatic Categorization of Ottoman Literary Texts by Poet and Time Period", Computer and Information Sciences II, 2011, pp. 51-57. doi:10.1007/978-1-4471-2155-8_6

[24] B. Mehta, B. Rajyagor., "Gujarati Poetry Classification Based On Emotions Using Deep Learning", International journal of Engineering Applied Sciences and Technology, 2021. Vol. 6, Issue 1, pp. 358-362

[25] C. Tanasescu, B. Paget, D. Inkpen., "Automatic Classification of Poetry by Meter and Rhyme", Association for the Advancement of Artificial Intelligence, 2016

[26] R. A. Deshmukh, S. Kore, N. Chavan, S. Gole, K. Adarsh., "Marathi Poem Classification using Machine Learning", Blue Eyes Intelligence Engineering & Sciences Publication, 2019. Vol. 8, Issue 2, pp. 2723-2727. DOI: 10.35940/ijrte.B1761.078219

[27] S. Ahmad , M. Zubair, F. Mazaed, S. Khan., "Classification of Poetry Text Into the Emotional States Using Deep Learning Technique", IEEE Access, DOI: 0.1109/ACCESS.2020.2987842, 2020, Vol. 8, pp. 73865-73878.

[28] A. Lou, D. Inkpen and C. asescu., "Multilabel Subject-based Classification of Poetry", Association for the Advancement of Artificial Intelligence, 2015.

[29] T. Peri-Polonijo, "The Levels Of Classification Oral Lyrical Poems Classification", Nar. umjet. 32/1, 1995, pp.55-67.

[30] V. Kesarwani, "Automatic Poetry Classification Using Natural Language Processing", Thesis, School of Electrical Engineering and Computer Science Faculty of Engineering, University of Ottawa, Canada, 2018.