International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(1s) 2026 pp. 1-13

https://doi.org/10.70917/ijcisim-2026-1964
Article

A Novel Approach of Multi-label classification for low resource Braj Language with integration of Synonyms and POS Tagging

Mukta M.Deshpande[1], Prafulla B. Bafna[2*]

[1]Research Scholar, Symbiosis Institute of Computer Studies and Research, Pune India.
[1]Assistant Professor, G.S. Moze College of Engineering, Balewadi, Pune India.
mukta_deshpande12@yahoo.com
[2]Assistant Professor, Symbiosis Institute of Computer Studies and Research, Pune India.

Abstract: - This research explores Braj language as one of the niche Indian low-resource languages for text analytics for the very first time. The research mainly focuses on entities synonym extraction, POS (part of speech tagging). Three feature extraction methods namely deep learning, transformer and count based are implemented for multi label classification tasks on the Holi Sagar Braj dataset. The best results are achieved with a combination of Linear SVM classification and TF-IDF with F1 score 0.88, precision 0.81, and recall 0.97. Results showed low efficiency in evaluating LSTM and IndicBERT models on the same dataset due to dialectal forms of the language challenges. The research on the Braj language for multi label classification advances in feature engineering and classification techniques for Indian regional language processing.
Keywords: Braj language, Multi label Classification, LSTM, IndicBERT, TF-IDF, synonyms, POS tagging

1. Introduction

Language is a means of communication that consists of a set of sounds and symbols used by people of a country or region for speaking and writing. The implementation of computational linguistics methods for text analytics has increased in recent years due to the volume of textual data. Artificial Intelligence (AI) aimed at making computers understand the words of a text corpus. But major developments in the field did not include all languages, Indian languages in particular remained far behind. From the context of training AI systems, the amount of data available for Indian languages, especially regional languages and dialects was less [2].

Indian regional languages are considered as poor languages because of their inadequate digital existence, poor corpora and pre-trained models [3].

India's federal governance requires regional languages for state administration, making it essential to develop techniques for extracting insights from these languages. Additionally, the increasing use of online digital applications highlights the need for NLP (Natural Language Processing) resources to support Indian regional languages.

In recent advancements, research has been carried out on computational resources and techniques to retrieve information from Hindi, Sanskrit and Indian vernacular languages but extensive research is yet to be performed on Braj language. Braj language from Mathura and the regions associated with Lord Krishna's childhood in the western regions of modern-day Uttar Pradesh [4] and is mostly associated with the worshipers of Lord Krishna [5].

Numerous challenges are associated with Processing low resource as well as dialectal languages. The main concern is because of lexical variation. In lexical variation a single idea can have several synonymous vocabularies. Nowadays, the language is insignificant as a spoken language but vastly significant in literature in the form of poems and songs composed for Lord Krishna. Braj language with a rich literary heritage spoken predominantly in northern India, faces challenges due to its limited linguistic resources [6]. The lack of appropriate datasets, pre-

trained language models, and natural language processing (NLP) tools reduces the potential of this language for study, teaching, and cultural preservation by preventing the development of effective NLP applications.

The primary step involves developing an understanding of the vocabulary. For this reason, the pre-processed text is represented as a collection of word frequencies or Bag of Words (BOW). Effective pre-processing approaches are required to clean noisy data, as it directly affects the classification algorithm performance. Good text data certifies enhanced learning models and accurate interpretation. Text pre-processing significantly impacts natural language processing applications. Effective pre-processing is essential for new text corpora, especially in morphologically rich languages [7].

Synonyms provide valuable semantic information, enabling the model to capture the nuances of meaning and recognize different word forms [8],

POS tagging provides important grammatical information which includes identification of part of speech of every word including noun, adjective and verb [9].

Due to the massive growth of digital content, automatically classifying text has become an important NLP task for organizing and handling information professionally [10].

Multi-class classification technique comprises more than two classes. Evaluation metrics are needed for comparison of various classification techniques.

Machine Learning and Deep Learning Approaches

Current studies in Natural Language Processing (NLP) utilizes deep learning methods namely Long Short Term Memory (LSTM) networks, Convolutional Neural Networks (CNN) for text classification. Features are extracted in CNN through convolution and pooling whereas LSTM uses long term word dependencies improving classification accuracy [11]. IndicNLP Suite consists of pre-trained models IndicFT and IndicBERT. These pre-trained models help process Indian languages efficiently [12]. Stanza is a Python toolkit which supports many human languages as well as it is suitable for many tasks [13].

Feature Engineering for Braj Language Text Classification

Machine learning algorithms need numerical inputs therefore, unstructured data should be transformed into numerical inputs with suitable encoding methods [14].

A huge number of words in the corpus are only syntactic units and convey no meaning, such words are called Stopwords. A high frequency of Stopwords affects an NLP systems performance during Information Retrieval because frequent words are weighted more than less frequent words [15].

Therefore, removing Stopwords is necessary to improve performance and reduce index size by 30% [16].

The limited availability of annotated datasets for Indian regional language poses a significant hurdle in training effective machine learning models. The lack of readily available tools and resources for Indian regional language, such as pre-trained language models, part-of-speech taggers, and word embeddings, requires the development of custom solutions [17-18].

To create numeric features based on word statistics the TF-IDF algorithm is used [19]. The algorithm considers the importance of a word using two mathematical terms TF and IDF.

The first equation calculates the term's word frequency in the text and the second computes the inverse frequency of the word, both the equations are combined together in the third equation to compute the statistical measure. The algorithm has shown excellent results for text classification of low level Indian languages such as Bangla , however it treats each language as a separate unit and does not cover the lexical nature of words [20], therefore, BERT (Bidirectional Encoder Representations from Transformers) used as a comparison approach to retain the semantic features while creating feature vectors. BERT is a transformer technique which extracts features based on context and its text position [21].

A POS (parts-of-speech) refers to the tag associated with Nouns, Verbs, Adjectives, Pronouns, etc. and these tags follow a traditional convention defined by tagsets [22].

Classification allows for the identification of different types of texts, such as classifying literature by genre or identifying authorship [23].

To conduct this research, comparison of three feature extraction techniques namely TF-IDF, LSTM, and IndicBERT conducted for multi-label text classification of Braj language.

This research work makes following significant contributions:

1. Multi-label classification is explored for Braj language.
2. Compared three different feature extraction techniques including count based, deep learning and transformer based for Braj language on Holi Sagar data set.
3. Contribution for Low resource Indian regional language processing.

The paper is organized in following sections. An exhaustive literature study on the research work is carried out in section 2. Section 3 describes the overall working methodology of the current research study. Results are explained in section 4. Section 5 concludes the research study.

2. Literature Review

It is vital to understand the background that gives direction to this study. This section discusses the past research related to text pre-processing, text mining, feature extraction, and training a text classifier.

Khurana, D et.al (2023) conducted a survey on Indian Language Text processing. They have identified challenges associated with Indian language processing. In natural language processing, understanding contextual words and terms is one of the key challenges when working with synonyms.

Desai, N. P., & Dabhi, V. K. (2021) in their research observed that Hindi wordnet has not sufficient synsets as compared to English. There is still scope of improvement in Hindi Wordnet. Hindi tools are not released in the open domain.

Choudhary, N. (2021) observed that India has many languages as well as dialects spoken all over the country. Bhojpuri, Awadhi are a few examples of Hindi dialects which are greatly different from Hindi. This difference needs distinct support for language processing and model development.

Braj language is mainly spoken in Mathura and some parts of the North region of India. Braj language is literary language, largely focusing on Krishna. Braj language has enhanced poetry, culture and music in the county (Upadhyay, M. et al., 2023).

At present, Braj is home to thousands of holy shrines and temples connected with Krishna (Taneja, L., 2023).

Minar, M. R., & Naher, J. (2018) studied current improvements in Deep Learning over the last few years.

Chai, C. P. (2023) discovers the advantages and drawbacks of several text pre-processing approaches. Text pre-processing significantly impacts natural language applications, improving accuracy and coverage. Effective pre-processing is essential for new text corpora, especially in morphologically rich languages.

Considering the present growth of online information in natural languages is multilingual. The Indian regional languages are low resource languages. Harish, B. S., & Rangan, R. K. (2020) examines the many methods and strategies that researchers have developed for processing regional languages in India.

Panjwani, R et.al (2018) provides an API (Application programming Interface) for accessing IndoWordNet using Python. Pyiwn a Python based API to access Indian language WordNets.

Dhar, A. et.al (2018) Conducted text classification on Bangla Text. Text classification is an effective technique for organizing and handling textual information.

Jang, B. et.al (2020) observed accurate results for sentence classification using the LSTM model and the CNN. LSTM models are more effective in classifying texts because they capture long term dependencies between words.

Kakwani, D et.al (2020) presented IndicNLP suite which is a huge corpora having Billions words along with IndicFT, IndicBERT. IndicNLP is one of the useful resources for the Indian research community.

Qi, P et.al (2020) introduced a python based open source Stanza which supports 66 human languages.

Potdar, K. et.al (2017) Since machine learning algorithms can only take numerical inputs, encoding strategies are required to convert these category variables into numerical values.

Improved results interpretation and effective learning models are provided by high-quality text data. A high-quality text corpus requires effective pre-processing techniques since real-world text data contains a lot of errors and anomalies. Pre-processing includes a number of document transformation and manipulation procedures prior to text classification Avasthi, S. et.al (2022).

Sahu, S. S., & Pal, S. (2022) examine how Stopwords affect the ability to retrieve four different Indian languages. Across the four Indian languages, Stopwords typically have a relatively minor impact on short documents when compared to long documents.

Kunchukuttan, A et.al (2020) presents IndicNLP corpus for 10 Indian languages also they shared pre-trained word embedding on these corpus which will be useful for performing various NLP operations.

Liu, Q. et.al (2018) Conducted research using Word2Vec to capture word semantics in order to enhance text feature extraction. Words with low TF-IDF values are eliminated, and a density clustering technique is used to group related words together. To improve accuracy, these clusters are incorporated into a vector space model.

Liang, M., & Niu, T. (2022) conducted study to enhance text classification by altering TF-IDF, employing Word2Vec embeddings, and integrating bidirectional LSTM with Text-CNN for feature extraction and prediction, hence improving sequence processing and accuracy.

Gomes, L. et.al (2023) performs text classification with BERT as feature extraction with six different classification algorithms.

BERT is a bidirectional language model that utilizes deep contextual information from both sides, enabling fine-tuning for various applications of natural language processing with minimal changes. It improves transfer learning effects for bidirectional designs, enhancing performance even in low-resource tasks Devlin, J. (2018).

Text classification is an important approach applied to arrange and derive insights from textual data. The classes are further categorized by identifying the text types of the content. Accuracy is the most important aspect to take into account when applying machine learning algorithms on a specific data set, Shah, K. et.al (2020).

Grandini, M. et al. (2020) describe multi-class classification as a machine learning technique working on more than two classes. Performance metrics are important for evaluating and comparing classification models.

Research on Braj language, a significant Western Hindi dialect, requires the usage of innovative linguistic methods such as synonym identification and part-of-speech tagging. These methods are essential for understanding the language's rich vocabulary and complex grammatical structures. Synonym identification shows a significant role in revealing the different meanings and contexts in which words are used, helping researchers create more accurate and comprehensive Braj language dictionaries.

Part-of-speech tagging is another critical technique for studying Braj language. By automatically classifying words into their grammatical categories, researchers can build detailed models of the language’s sentence structure. This helps in analysing the language’s unique verb conjugations, idiomatic expressions, and overall syntactic patterns. Understanding these elements is crucial for a deeper exploration of Braj language’s grammar and usage.

The reason for selecting classification tasks in this context is that they provide an effective way to automate the organization of Braj language texts. By automating these processes, researchers can save time and improve the accuracy of their analyses.

Furthermore, classification tasks make it easier to manage and preserve Braj language literary heritage. Organizing texts through classification helps make them more accessible to researchers, and language enthusiasts.

3. Research Methodology

In order to overcome linguistic and contextual issues specific to Indian languages, this study attempts to create a strong multi-class classifier for the classification of Braj language data, which includes sentences relating to Krishna, Radha, and Holi. The methodology is divided into five main phases, pre-processing and data creation, synonym identification, sentence labelling and part-of-speech (POS) tagging, assigning labels to entities, and classifier building. In the classification challenge, three feature sets TF-IDF, LSTM embeddings, and IndicBERT embeddings were used to evaluate many algorithms, including Linear SVM, Logistic Regression, Random Forest, and KNN. In order to determine the optimal method and feature set combination, the evaluation concentrated on

precision, F1 Score and recall for five classes. Following Figure 1 explains the methodology followed for conducting this research work.

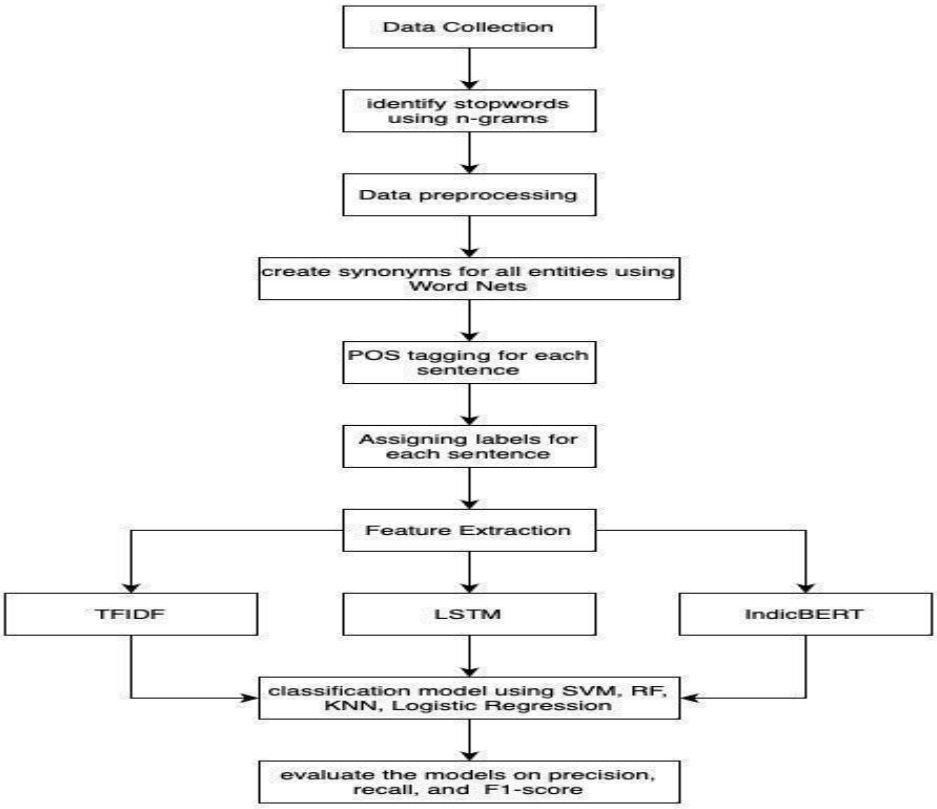


Figure 1: Overview of the research methodology.

3.1 Data creation and pre-processing:

Text pre-processing involves cleaning and transforming raw text for further analysis. For this study, it comprises removing punctuation and Stopwords.

3.2. Identifying synonyms for Krishna, Radha, and Holi:

The corpus comprises words that refer to three common entities- Krishna, Radha, and Holi. To build an NLP system that identifies words from the corpus it is important to understand their meaning and the manner of their combination in sentences to convey meaning. Following Figure 2 shows the proposed Algorithm For synonym extraction and POS tagging on Braj Language Holi Sagar Dataset. From the Holi Sagar data set we have identified 22 synonyms for Krishna, 10 synonyms for Radha and 19 synonyms for Holi. These synonyms are presented as word clouds in Figure 5.

Algorithm1. Representing text corpus as a categorical dataframe

Input: preprocessed corpus T

Output: df

1. Generate list of synonyms Lk, Lr, Lh using pyiwn for Krishna, Radha, and Holi
2. Create Lt by Sentence Tokenization of T
3. Create custom Ldk, Ldr for discourse matching
4. df['sentences'] <- Lt
5. For row in df.iteritems() do

```
Create POS tags for row['sentences']
If NNP, NNC, and NN labels in POS tags then
   noun <- POS['text']
   If noun in Lk then
      If noun in Lr then row['entity'] <- Kishna and Radha
       Else row['entity'] <- Krishna
    End if
    If noun in Lr then row['entity'] <- Radha

    If noun in Lh then row['entity'] <- Holi
    Else row['entity'] <- None
End if
If row['entity'] is None then
  For sent in row['entity'] do
    If sent in Ldk then entity['row'] <- Krishna
    If sent in Ldr then entity['row'] <- Radha
  End for
```

6. End for
7. df['label'] <- OrdinalEncoding(df['entity'])
8. Return df

Figure 2 : Proposed Algorithm For synonym extraction and POS tagging.

3.3 POS Tagging for each sentences

There exists no specific resource to identify POS tags for Braj because no defined tagset exists for the language (However, Stanza generated the Proper Nouns (NNP), Common Nouns (NN), and Personal (PRP) tags with good accuracy for the Hindi language. To map the identified POS tags with the identified entities an independent list of synonyms was created for each entity using pyiwn. Pyiwn is a Word Net approach developed by researchers at the Indian Institute of Technology, Bombay to create synsets or a wordlist in Hindi for a given word. The overall approach to map each sentence with the three entities lexically is summarized in Figure 3 below.

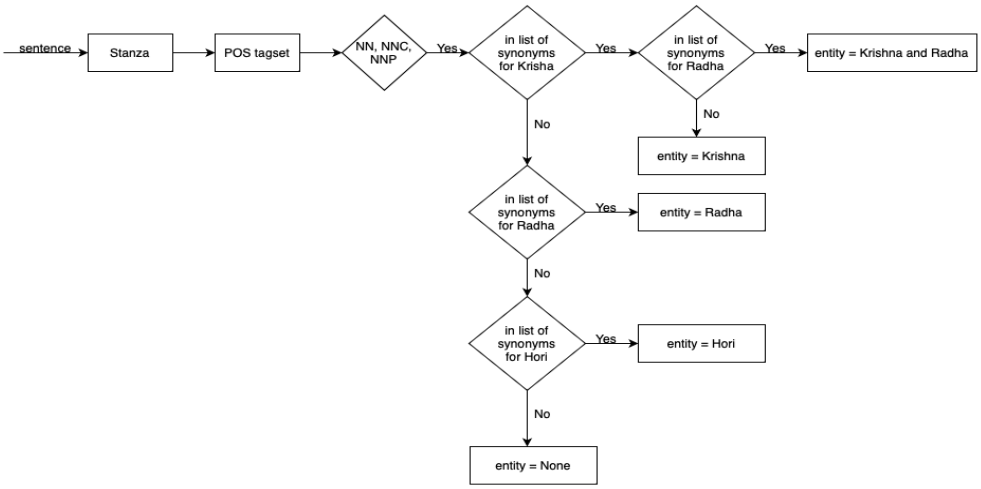


Figure 3: Approach to map each sentence with list of synonyms based on meaning.

3.4 Assigning Labels to entities:

To train a supervised Machine Learning algorithm to create a classifier it is essential to signify the text corpus as a Data frame with one column containing numeric labels. Like the English language, the Devanagari script follows a characteristic grammar, however after pre-processing the corpus lacks punctuations and splitting it into sentences is a challenge. A solution is to tokenize the pre-processed corpus using the IndicNLP Sentence Tokenizer and transform the sentences as an unlabeled data frame.

3.5 Developing a multi-class classifier

In this section, supervised Machine Learning algorithms are applied on categorical data to classify each sentence into one of the entities. A classic Machine Learning algorithm requires numeric features as inputs therefore it is important to transform the 'sentence' column into numeric features. The final stage involved developing a multi-class classifier to categorize sentences effectively. Additional linguistic features, such as POS tags and keyword frequencies, were included to enrich the feature set. Various machine learning and deep learning algorithms were explored, including Logistic Regression and Random Forest for baseline comparisons and neural network architectures like BERT and LSTM for advanced modelling. The models were evaluated using metrics such as accuracy, F1-score, recall and precision. While confusion matrices were provided for visualizations.

4. Result Analysis

The pre-processed corpus is shown in Figure 4.

```
Processed Text:
आज बिरज होरी रसिया
उतते आये कुँवर कन्हैया इतते राधा गोरी रसिया
उड़त गुलाल अबीर कुमकुमा केशर गागर ढोरी रसिया
बाजत ताल मृदंग बांसुरी और नगारे जोरी रसिया
कृष्णजीवन लच्छीराम प्रभु सौं फगुवा लियौ भर झोरी रसिया

सजनी भागन फागुन आयो खेलूँगी श्याम सँग जाय

खेलूँ आप खिलाऊँ लाल मुख पे मलूँ गुलाल
वाने भिजोई मेरी फूलन अँगिया भिजोऊँ वाकी पाग
चोबा चन्दन अतर अरगजा अबीर गुलाल उड़ाय
बरज रही बरज्यो नहिं मान्यों हियरा उठ्यो अनुराग
फेंट गुलाल हाथ पिचकारी करत अनौखे ख्याल
जो खेलो तौ सूधे खेलौ मारूँगी गुलचा गाल
कृष्ण जीवन लच्छी राम प्रभु सौं मानूँगी भाग सुहाग

रसिया होरी मेरे लग जायेगी मत मारै दृगन चोट
```

Figure 4: Sample Dataset for Braj Classification

After mapping each sentence with the three entities based on the meaning of words some sentences remained unclassified and required discourse matching. To perform discourse matching for each entity custom lists of words were curated (Table 1) after manually identifying the pronouns referring to the entities. The final list of curated synonyms for each entity are depicted in Figure 5 by word clouds.

Table 1: curated list of words for discourse matching.

Entity Name	Words
Krishna	तेरे, तेरो
Radha	मेरी, मेरे, मेरो, तेरी

Figure 5: Synonyms generated for the three categories.

Each entity is assigned an integer label by Ordinal Encoding. The count of sentences in each category are depicted in Figure 6. We can observe there were 651 sentences present for Krishna, 244 for Radha and 372 sentences for describing Holi. There were 119 sentences Combined for Krishna and Radha.

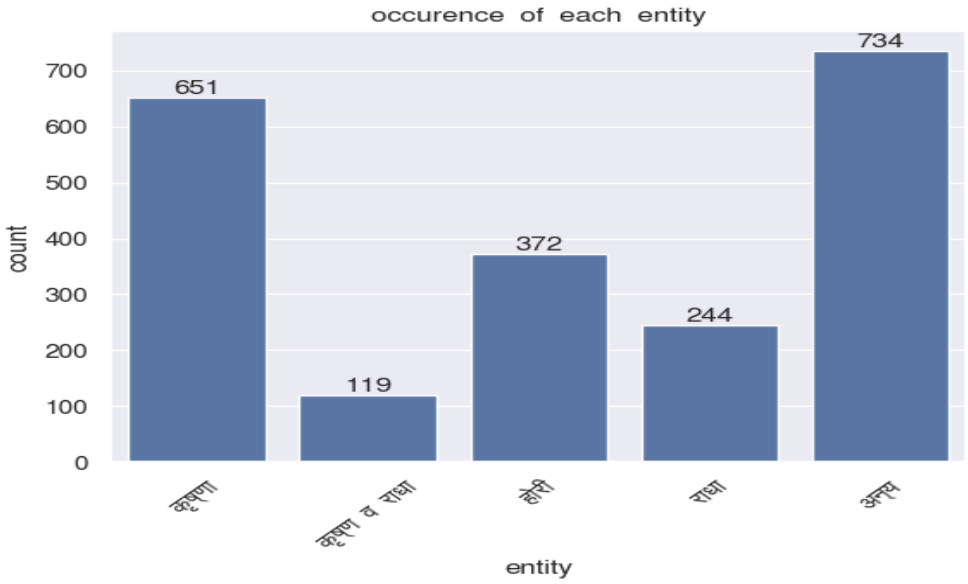


Figure 6: Classification of each sentence into 5 entities.

As the next step each sentence is mapped to the identified entities as explained in the preceding section.

	sentence	entity	label
0	आज बिरज होरी रसिया	कृष्णा	2.0
1	उतते आये कुँवर कन्हैया इतते राधा गोरी रसिया	कृष्ण व राधा	1.0
2	उड़त गुलाल अबीर कुमकुमा केशर गागर ढोरी रसिया	कृष्णा	2.0
3	बाजत ताल मृदंग बांसुरी और नगारे जोरी रसिया	कृष्णा	2.0
4	कृष्णजीवन लच्छीराम प्रभु सौं फगुवा लियौ भर झोर...	कृष्णा	2.0
...	...	...	...
2115	नंदगाम आये सखा सब बरसाने बाम	कृष्णा	2.0
2116	गहवरवन और खोरसाँकरी कुँज कुटी निज धाम	अन्य	0.0
2117	राधे कूँ मनमोहन कीये मोहन बनाय दिये नारि	कृष्ण व राधा	1.0
2118	हाथन मेंहँदी पाँय महावर बेंदी लगाय दई भाल	अन्य	0.0
2119	कहत दास नवनीत पियारौ जपूँ तिहारौ नाम	अन्य	0.0

Figure 7: Text corpus represented as categorical data

As observed from Figure 7, the created categories are imbalanced. The dataset can be interpreted as a multi-majority class dataset or a dataset with multiple majority classes and a single minority class. Such datasets pose challenges for Machine Learning algorithms because the decision boundaries between many classes are considered complex and challenging to teach and affect the performance of classifiers. Most contemporary research focuses on addressing binary unbalancing and lacks a specialized approach to address multi class unbalancing. In this research supervised Machine Learning algorithms are trained over the imbalanced dataset as explained below.

The typical structure of a Machine Learning algorithm includes a vectorizer and a classifier. In the current implementation the four algorithms serve as the classifier while the generated features serve as the vectorizer shown in Table 2. For implementation using the TF-IDF features. For using the features of IndicBERT and LSTM no such structure was created, the classifier was trained on the generated features instead.

Table 2: classifiers using TF-IDF, LSTM and IndicBERT as feature Extraction

Algorithm	**Class**	**TF-IDF Features**			**LSTM Features**			**IndicBERT Features**		
		Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Linear SVM	0	**0.81**	**0.97**	**0.88**	0.82	0.9	0.86	0.6	0.02	0.04
	1	**0.76**	**0.62**	**0.68**	0.85	0.52	0.65	0.2	0.1	0.13
	2	**0.9**	**0.8**	**0.84**	0.85	0.8	0.82	0.4	0.01	0.03
	3	**0.84**	**0.71**	**0.77**	0.76	0.69	0.72	0.11	0.87	0.2
	4	0.82	0.83	0.83	**0.81**	**0.9**	**0.85**	0.39	0.27	0.32
Logistic Regression	0	0.59	0.99	0.74	**0.81**	**0.9**	**0.86**	0.33	0.92	0.49
	1	1	0.33	0.5	**0.92**	**0.52**	**0.67**	0	0	0
	2	0.87	0.73	0.8	**0.83**	**0.82**	**0.82**	0.59	0.19	0.29
	3	1	0.18	0.3	**0.76**	**0.62**	**0.68**	0	0	0
	4	0.88	0.69	0.77	**0.83**	**0.87**	**0.85**	0	0	0
Random Forest	0	**0.83**	**0.99**	**0.91**	0.84	0.9	0.87	0.39	0.7	0.51
	1	**0.8**	**0.57**	**0.77**	0.5	0.57	0.53	1	0.14	0.25
	2	**0.91**	**0.82**	**0.87**	0.87	0.79	0.83	0.53	0.54	0.53
	3	**0.88**	**0.67**	**0.76**	0.75	0.67	0.71	0.33	0.02	0.04
	4	**0.86**	**0.93**	**0.89**	0.83	0.89	0.86	0.33	0.1	0.15
KNN	0	0.64	0.61	0.62	**0.82**	**0.9**	**0.86**	0.41	0.44	0.42
	1	0.6	0.57	0.59	**0.76**	**0.62**	**0.68**	0.32	0.38	0.35
	2	0.64	0.65	0.65	**0.83**	**0.8**	**0.81**	0.48	0.47	0.48
	3	0.42	0.31	0.36	**0.77**	**0.67**	**0.71**	0.14	0.11	0.12

	4	0.49	0.62	0.55	**0.83**	**0.87**	**0.85**	0.26	0.25	0.26

To visualize and summarize the performance of the classifiers a confusion matrix are shown in Figures 8, 9, and 10 for each classifier.

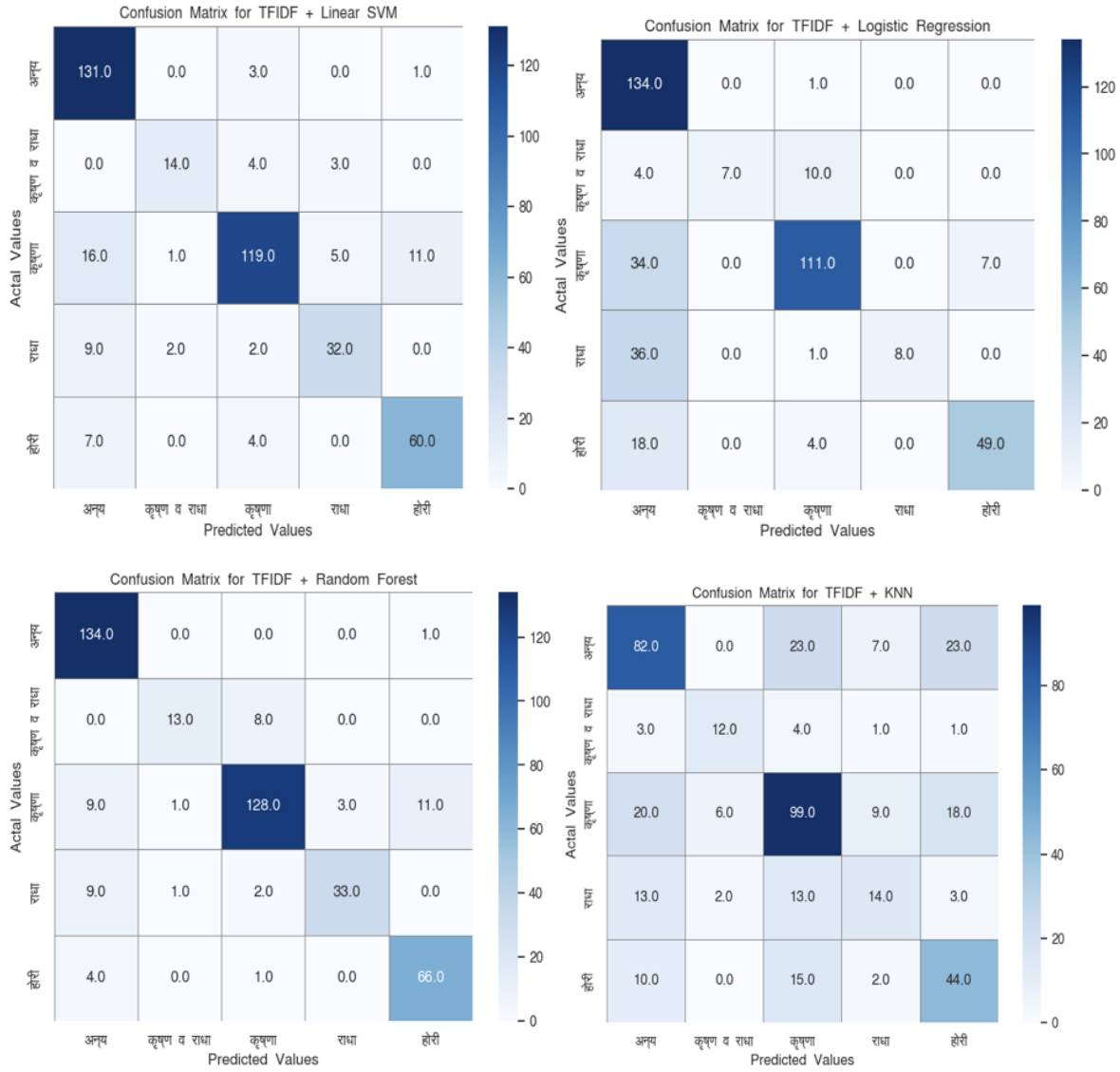


Figure 8: Confusion matrix of each classification algorithm using TF-IDF features.

TF-IDF proved to be the most consistent feature set across all algorithms. Random Forest emerged as the best-performing algorithm with TF-IDF, achieving the highest F1-scores for most classes. Linear SVM also demonstrated strong results, particularly for Classes 0, 2, and 4, with F1-scores exceeding 0.8. Logistic Regression struggled with some classes (e.g., Class 3, F1 = 0.3), indicating its limited ability to generalize across certain sentence types. Overall, TF-IDF provided a reliable foundation for classification tasks.

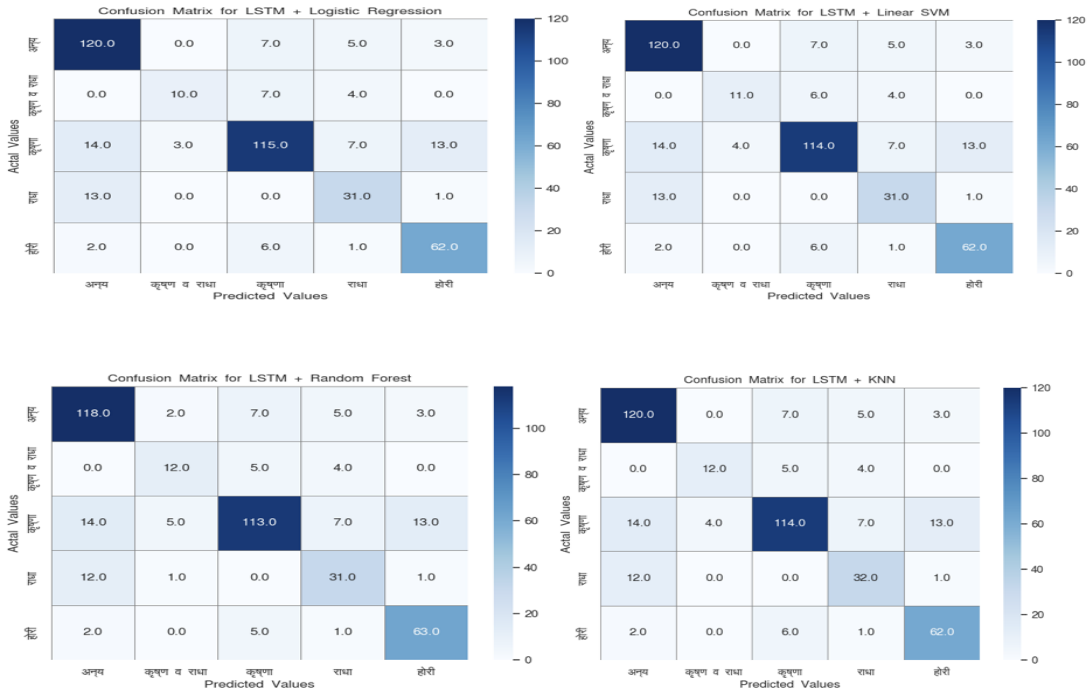
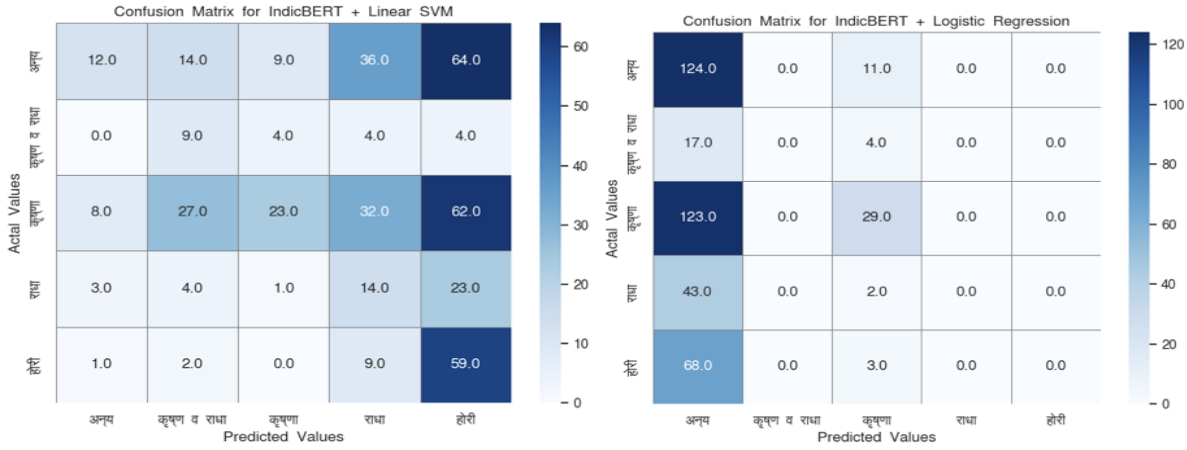


Figure 9: Confusion matrix of each classification algorithm using LSTM features.

The use of LSTM embeddings improved overall performance compared to TF-IDF for most algorithms. Random Forest and SVM demonstrated similar effectiveness, with notable gains in precision and recall for Class 4 (F1 = 0.86 for Random Forest). Logistic Regression showed balanced performance across all classes, with improved F1-scores compared to TF-IDF. The results suggest that LSTM embeddings capture richer contextual information, enabling better classification performance.



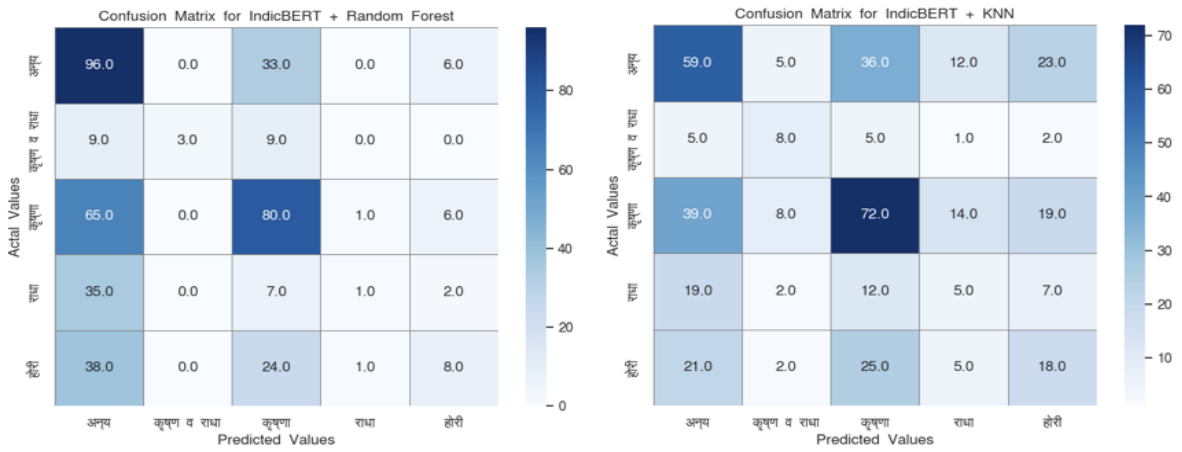


Figure10: Confusion matrix of each classification algorithm using IndicBERT features.

IndicBERT embeddings underperformed significantly across all algorithms. Linear SVM and Logistic Regression produced particularly low precision, recall, and F1-scores, with several classes (e.g., Class 1 and Class 3) failing entirely to be classified (F1 = 0). Random Forest performed slightly better but still failed to achieve satisfactory results, with F1-scores below 0.6 for most classes.

5. Conclusion

Classification research on Braj language mythological texts enhances the global recognition and appreciation of Braj cultural heritage, positioning Braj language mythology as a valuable contribution to world literature and religious studies. The analysis underscores the importance of feature engineering and algorithm selection in multi-class classification tasks. While TF-IDF and LSTM embeddings demonstrated strong performance, IndicBERT embeddings showed significant limitations in this context. Random Forest, when paired with LSTM features, proved to be the most promising combination for developing a reliable multi-class classifier. Future work for this study will be exploring new feature extraction techniques to improve performance.

Conflict of Interest: The authors declare that they have no conflicts of interest in relation to the work presented in this manuscript.

Author Contribution: All authors contributed equally in the present manuscript.

Acknowledgement: Not applicable.

References

1. Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: state of the art, current trends and challenges. Multimedia tools and applications, 82(3), 3713-3744.
2. Desai, N. P., & Dabhi, V. K. (2021). Taxonomic survey of Hindi Language NLP systems. arXiv preprint arXiv: 2102.00214.
3. Choudhary, N. (2021). LDC-IL: The Indian repository of resources for language technology. Language Resources and Evaluation, 55(3), 855-867.
4. Upadhyay, M., Upadhyay, A. K., & Kumari, R. (2023). Contribution of Language and Literature in the Creation of Culture. Boletin de Literatura Oral-The Literary Journal, 10(1), 3252-3255.
5. Taneja, L. (2023). Losing and Finding Braj: Commodification and Entrepreneurship in the Sacred Land of Krishna. Religions, 14(5), 643.
6. Minar M R and Naher J. 2018. Recent advances in deep learning: An overview. Preprint gr- qc/180708169
7. Chai, C. P. (2023). Comparison of text pre-processing methods. Natural Language Engineering, 29(3), 509-553.
8. Harish, B. S., & Rangan, R. K. (2020). A comprehensive survey on Indian regional language processing. SN Applied Sciences, 2(7), 1204.
9. Panjwani, Ritesh and Kanojia, Diptesh and Bhatacharyya, Pushpak. 2018. Pyiwn: A Python based API to access Indian Language WordNets. In proceedings of the Global WordNet Conference
10. Dhar, A., Dash, N.S., Roy, K. 2018. Application of TF-IDF Feature for Categorizing Documents of Online Bangla Web Text Corpus. Published In: Bhateja, V., Coello, C., Satapathy, S., Pattnaik, P. (eds) Intelligent Engineering Informatics. Advances in Intelligent Systems and Computing

11. Jang, B., Kim, M., Harerimana, G., Kang, S. U., & Kim, J. W. (2020). Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism. Applied Sciences, 10(17), 5841.
12. Kakwani, D., Kunchukuttan, A., Golla, S., Gokul, N. C., Bhattacharyya, A., Khapra, M. M., & Kumar, P. (2020, November). IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 4948-4961).
13. Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020). Stanza: A Python natural language processing toolkit for many human languages. arXiv preprint arXiv:2003.07082.
14. Potdar, K., Pardawala, T. S., & Pai, C. D. (2017). A comparative study of categorical variable encoding techniques for neural network classifiers. International journal of computer applications, 175(4), 7-9.
15. Avasthi, S., Chauhan, R., & Acharjya, D. P. (2022). Significance of preprocessing techniques on text classification over hindi and english short texts. In Applications of Artificial Intelligence and Machine Learning: Select Proceedings of ICAAAIML 2021 (pp. 743-751). Singapore: Springer Nature Singapore.
16. Sahu, S. S., & Pal, S. (2022). Effect of Stopwords in Indian language IR. Sādhanā, 47(1), 17
17. Kunchukuttan, A., Kakwani, D., Golla, S., Bhattacharyya, A., Khapra, M. M., & Kumar, P. (2020). Ai4 Bharat-indic nlp corpus: Monolingual corpora and word embeddings for Indic languages. arXiv preprint arXiv: 2005.00085.
18. Liu, Q., Wang, J., Zhang, D., Yang, Y., & Wang, N. (2018, December). Text features extraction based on TF-IDF associating semantic. In 2018 IEEE 4th international conference on computer and communications (ICCC) (pp. 2338-2343). IEEE.
19. Liang, M., & Niu, T. (2022). Research on text classification techniques based on improved TF-IDF algorithm and LSTM inputs. Procedia Computer Science, 208, 460-470.
20. Gomes, L., da Silva Torres, R., & Côrtes, M. L. (2023). BERT-and TF-IDF-based feature extraction for long-lived bug prediction in FLOSS: a comparative study. Information and Software Technology, 160, 107217.
21. Jacob Devlin and Ming-Wei Chang and Kenton Lee and Kristina Toutanov. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv: 1810.04805
22. Shah, K., Patel, H., Sanghvi, D., & Shah, M. (2020). A comparative analysis of logistic regression, random forest and KNN models for the text classification. Augmented Human Research, 5(1), 12.
23. Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for multi-class classification: an overview. arXiv preprint arXiv:2008.05756.