Text-Based Emotion Classifier Using Machine Learning Methods

G.Vyshanavi
School of Computer Science and Engineering
VIT-AP University
vyshnavi.20bci7004@vitap.ac.in

M.Jahnavi
School of Computer Science and Engineering
VIT-AP University
chandrika.20bcd7044@vitap.ac.in

SHK. Raghavendra
School of Computer Science and Engineering
VIT-AP University
raghavendra.21mis7106@vitapstudent.ac.in

K.Karthik
School of Computer Science and Engineering
VIT-AP University
karthik.21mic7164@vitapstudent.ac.in

M.Wasim Khan
School of Computer Science and Engineering
VIT-AP University
wasim.21mis7125@vitapstudent.ac.in

Abstract: A vast range of textual content in several Indian languages is available in digital format in today's digital landscape, which poses a challenge for efficient retrieval and categorization. This problem is solved by text classification, a branch of text mining that groups documents according to their content. This research examines text categorization methods used for content in Indian languages while taking into account the unique difficulties in natural language processing. The study shows that a number of supervised learning algorithms perform well on tasks involving text classification, including Naive Bayes, Support Vector Machines, Artificial Neural Networks, and N-gram. The study highlights how important text classification is to the administration and structuring of large amounts of textual data.

Keywords: Support Vector Machine (SVM), Natural Language Processing (NLP), Naive Bayes Algorithm, Category, and Supervised Learning Methods.

INTRODUCTION

The exponential growth of the World Wide Web has led to an extensive accumulation of data, primarily in the form of text. However, the abundance of information poses a challenge in identifying pertinent knowledge. Addressing this challenge, text classification categorizes a set of input documents into predefined classes, enabling efficient organization and information retrieval [1]. Text classification is an essential text mining approach that is used in many different applications, including web page hierarchical classification, document indexing, and organization. Through the automation of the classification process, text classification provides significant advantages over manual methods, including enhanced speed and efficiency.

Language serves as the primary means for both written and spoken communication. Web-based content is available in multiple languages due to the usage of Unicode encoding, which brings the complexities of natural language processing to text classification. Thus, text categorization includes natural language processing as well as text mining. The process entails combining machine learning (ML) and information retrieval (IR) technologies to give documents keywords and classify them into distinct groups.. ML algorithms facilitate automatic categorization, while IR techniques represent text through various features.

In recent times, the expansion of the internet in India has resulted in a significant increase in the creation of digital content in Indian languages. This has resulted in an increased demand for text classification in Indian languages to efficiently organize and retrieve this vast amount of data. Text classification in Indian languages has the potential to enable multi-lingual communication, preserve cultural heritage, and improve access to information for non-English speaking populations.

Despite the obstacles presented by the diverse Indian language landscape, notable advancements have been achieved in the creation of text classifiers for Indian languages. However, there is still a need for further research in this area to address challenges such as the lack of annotated datasets and standardized language resources.

This paper intends to offer a comprehensive examination of the different strategies employed in text classification in Indian languages and emphasize the particular research conducted in this context. We will explore the techniques used for training classifiers with limited annotated data and discuss the effectiveness of transfer learning in this context. Additionally,

We will explore the applications of text classification in Indian languages, encompassing sentiment analysis, topic modeling, and document classification. Finally, we will discuss the potential impact of text classification in Indian languages on the digital landscape of India and the opportunities it presents for improving access to information and communication in Indian languages.This focuses on analyzing the application of text classifiers in different Indian languages. An outline of the procedures in the text classification process is given in Section II. In Section III, the many methods used for text classification are discussed, and the work done specifically for Indian languages is highlighted.

LITERATURE SURVEY

CNN_Text_Word2vec, a microblog emotion categorization model proposed by Dongliang Xu et al. [1], used a convolutional neural network (CNN) for feature extraction. The model achieved good classification results and outperformed other methods such as Regarding accuracy of emotional classification, SVM, RNN, and LSTM are superior. However, one limitation of the model was the improper ranking between the extracted features. Our proposed work incorporates an enhanced feature ranking algorithm to address the issue of improper feature ranking, thereby enhancing the classification accuracy.

Using a set of fuzzy rules, Srishti Vashishtha et al. [2] created a sentiment analysis system for social media messages. Their method combined word sense disambiguation and natural language processing (NLP) approaches across several lexicons and datasets. The fuzzy system solved linguistic issues and produced reliable sentiment values. It was built on a novel unsupervised nine fuzzy rule-based system. Although the strategy performed better than other cutting-edge techniques, its high mistake rate resulted in incorrect class fixation. Our work produces more accurate sentiment values by reducing the high error rate seen in the fuzzy rule-based system through the use of a novel supervised learning method.

A multi-label maximum entropy (MME) model was presented by Jun Li et al. [3] for the purpose of classifying user emotions in brief messages. Based on user valence scores and several emotion labels, the MME model produced extensive features. Using produced lexicons, the approach successfully detected entities and delivered corresponding social feelings. While the method was effective in classifying social emotions over sparse features, it had issues with overfitting. In our study, we implement a regularization technique to mitigate overfitting, ensuring a more robust and generalizable model for user emotion classification.

A system that merged distributed word representations (DWRs) on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) variations with weighted attentive pooling (WAP) was created by Fazeel Abid et al. [4]. This was done using a weighted mechanism. The plan addressed terms that are not in the vocabulary (OOV) as well as syntactic and semantic regularities. According to the experimental investigation, the scheme's accuracy rate was 89.67%. Its insufficient feature extraction, which resulted in analytical errors, was a drawback. Our proposed scheme addresses the limitation of inadequate feature extraction through the incorporation of advanced word representation models, resulting in improved accuracy.

For sentiment analysis of Chinese microblogging systems, Peng Wu et al. [5] developed an Ortony-Clore-Collins (OCC) model and an institutionalization technique based on convolutional neural networks (CNNs). The system surpassed other state-of-the-art methods in terms of classification and recognition performance by combining emotion cognition with deep learning. On the other hand, the technique has a problem with microblog sentiment classification due to its intricacy. Our work simplifies the complexity issue associated with microblog sentiment classification by optimizing the OCC model and CNN-based institutionalization method, leading to improved classification performance.

To determine the degree of extremist attitude, Muhammad Asif et al. [6] used sentiment analysis on multilingual textual data from social media. The program successfully extracted extreme sentiment from multilingual data, achieving an 82% accuracy rate overall. The plan performed better in terms of dependability and scalability than previous methods. On the other hand, it performed worse when it came to multimodal sentiment recognition. Our work aims to improve multimodal sentiment identification by integrating the most recent deep learning techniques, which will lead to increased reliability and scalability.

PROPOSED WORK

Every paragraph needs to be nested. Every paragraph needs to be justified, both to the left and to the right.

Text Classification Process

There are various sub-phases in the text classification process, and each is essential to getting reliable classification results. The fundamental steps involved in text classification are shown in Figure 1, and they are as follows: gathering data, pre-processing, feature extraction, feature selection, creating a classifier, and performance evaluation [1] [2]. The following describes each sub-phase's goal and significance:

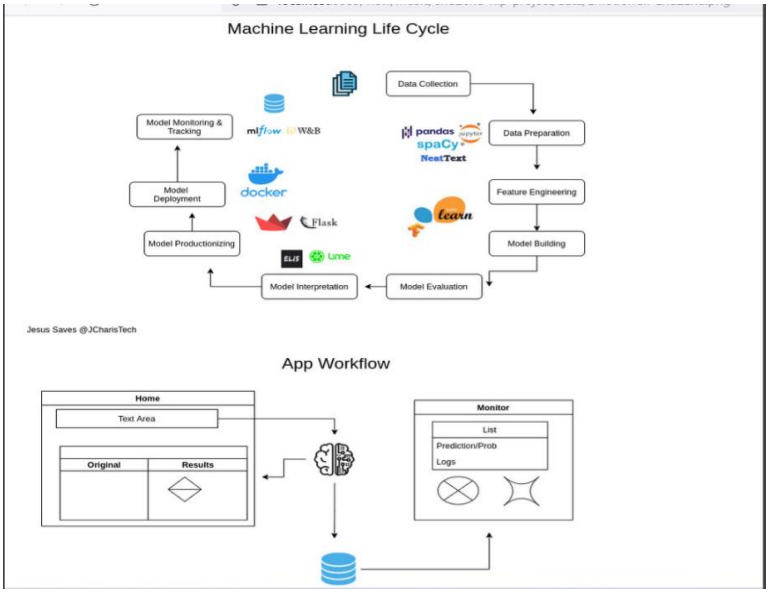
Data Collection

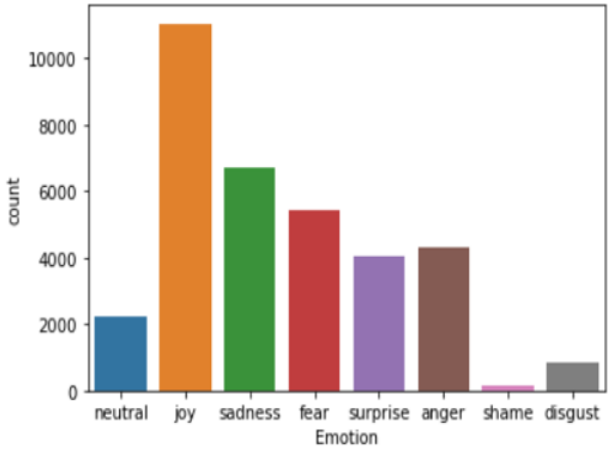
As part of the classification process, a corpus of documents in several formats, including.html,.pdf,.doc, and web content, are gathered. The classifier is trained and tested using these documents.

Pre-Processing

Pre-processing entails converting the text documents into an understandable Word format. This stage, which includes the following standard procedures, gets the documents ready for additional text classification processing:

One of the most important steps in the text categorization pre-processing stage is tokenization. It entails segmenting a text document into discrete words or tokens, which are subsequently utilized as classification characteristics. Natural language processing (NLP) methods like regular expressions, which can recognize special characters and word boundaries, are commonly used for tokenization. Additionally, tokenization can also take into account multi-word expressions or n-grams, which involve grouping together adjacent words to capture their combined meaning.





Eliminating stop words: Stop words are frequently used terms in a language that have little semantic significance. Eliminating stop words can decrease the feature space's dimensionality and increase the classification model's accuracy.. However, in some cases, removing stop words may not be appropriate, especially if the stop words carry important information for classification. Because of this, it's critical to carefully evaluate how relevant stop words are to each unique text categorization task.

Stemming words: Stemming is a technique used to reduce words to their root form or stem. This is important in text classification because it can help to reduce the number of features and prevent sparsity in the feature space. There are several algorithms available for stemming, including the

Porter stemming algorithm and the Snowball stemming algorithm. However, stemming can sometimes result in the loss of important information or introduce errors, so it is important to evaluate the effectiveness of the stemming algorithm for each specific task.

Feature Extraction

One pre-processing method used to make documents easier to handle and less complex is feature extraction. The documents are converted to document vectors in this stage from their full-text version. The count vectorizer model (CVM), in which documents are represented by vectors of words, is the most often used document representation. Nevertheless, there are drawbacks to CVM, such as its high dimensionality, loss of semantic links between phrases, and loss of correlation between neighboring words. Term weighting techniques can be used to assign the terms the proper weights in order to overcome these problems.

Feature Selection

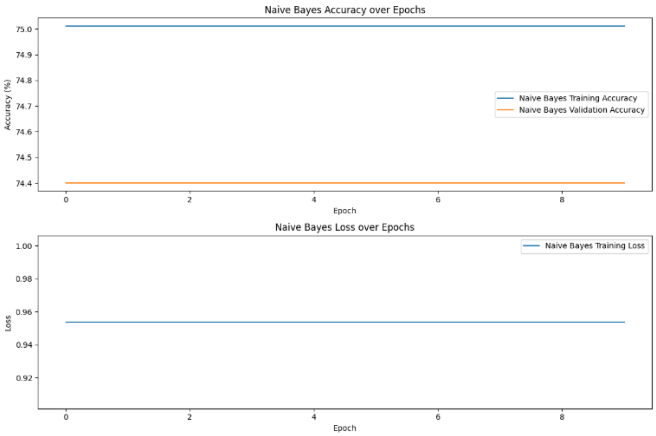
Feature selection is an important step in text categorization, following pre-processing and feature extraction.

Its goal is to build a vector space that enhances the text classifier's accuracy, efficiency, and scalability. Choosing a portion of the characteristics from the original papers is known as feature selection. In order to complete this process, the words with the greatest scores in relation to a predefined word importance metric are identified. Text classification has significant challenges due to the large dimensionality of the feature space; hence, a variety of feature evaluation metrics are employed, such as information gain (IG), term frequency, Chi-square, odds ratio, weight of evidence, mutual information, and Gini index [1].

Classification

The classification phase involves automatically categorizing documents into predefined categories. There are three main methods for document classification: unsupervised, supervised, and semi-supervised. The field of automatic text classification has advanced significantly in recent years, especially with regard to machine learning techniques like the Bayes classifier, logistic regression, support vector machines (SVMs), random forest, and gradient boosting.

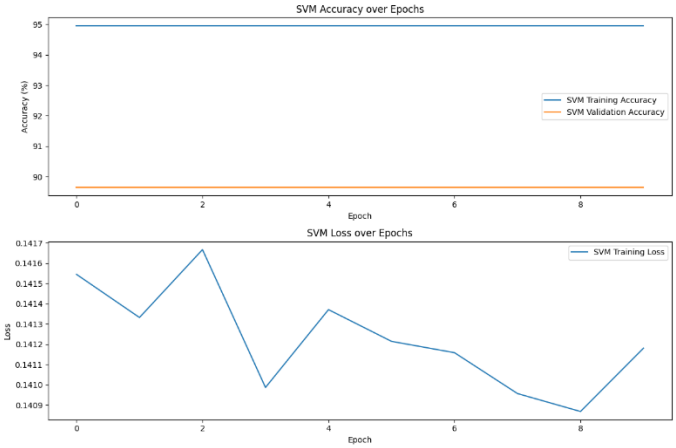
1)Navie Bayes: A straightforward probabilistic classifier known as Naive Bayes uses strong independence assumptions to apply the Bayes theorem. It has been widely used for text classification due to its effectiveness in dealing with large vocabularies typically found in text data. Naive Bayes models work well for text classification because they consider words or vocabularies as evidence. NB has been used for document classification in Indian languages.



(1)

2)**SVM:** Vapnik proposed the statistical classification technique known as SVM. It looks for a decision surface where training data points may be divided into two classes, then uses the support vectors to guide its decisions. SVM is regarded as one of the top text classification techniques and has been applied extensively in a number of research projects.Because Support Vector Machines (SVM) can handle high-dimensional data with small sample sizes, they are widely utilized in the text classification sector. By identifying the hyperplane that optimizes the margin between the two classes, SVMs have demonstrated their efficacy in effectively dividing the data points into two classes.

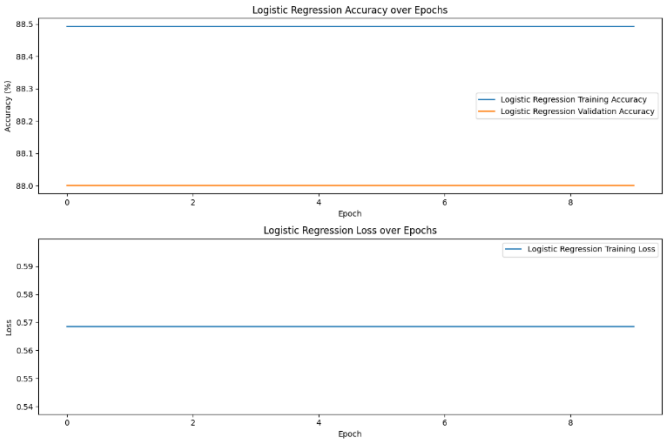
When it comes to text categorization, Support Vector Machines (SVM) can be used to train a model that divides documents into groups according to the presence or absence of particular keywords or attributes. By identifying the key characteristics that set each class apart, the SVM model gains th e ability to weigh each feature according to its significance. During the classification stage, the SVM model uses these weights to assign a document to the most appropriate category. SVM outperforms other text classification jobs using classification algorithms. Sentiment analysis, topic modeling, and document categorization are just a few of the text classification tasks for which SVM has been applied. To further increase classification accuracy, SVM has also been used in conjunction with other strategies like feature selection and ensemble approaches.



(2)

*3)**Logistic Regression:*** An A statistical technique called logistic regression is used to model the relationship between one or more independent variables and a binary dependent variable, or a variable with just two possible values, typically represented as 0 and 1. Regression analysis of this kind is frequently applied in predictive modeling and machine learning.By using the values of the independent variables, logistic regression seeks to determine the likelihood that a given observation is a member of a particular class. A logistic function, sometimes referred to as a sigmoid function, is the result of the logistic regression model. It maps any real-valued input to a value between 0 and 1.
The logistical role requires
the form:
$p(x) = 1 / (1 + e^{(-z)})$

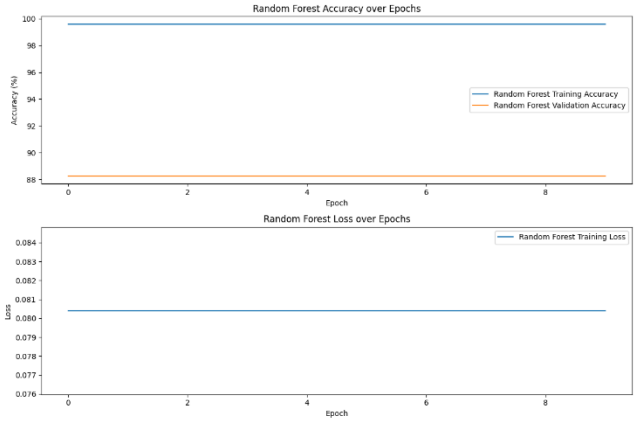
where z is a linear combination of the independent variables and their corresponding weights, x is a vector of independent variables, and p(x) is the probability that the dependent variable will be 1.



(3)

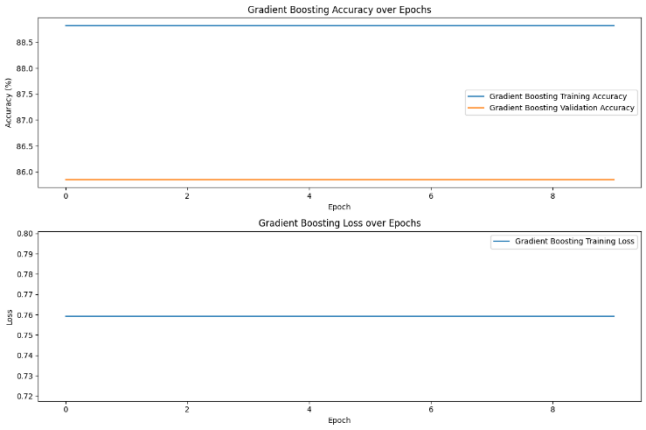
4) Random Forest: By mixing several decision trees, Random Forest is an ensemble learning technique that improves prediction accuracy and reduces overfitting. During training, it builds a large number of decision trees, and for classification tasks, it outputs the class mode. After being trained on a portion of the dataset, each tree adds to the ultimate result during prediction. This method is effective in handling large feature sets, and it provides a feature importance measure, highlighting the significance of each feature.

Random Forest is robust and applicable in text classification, particularly beneficial when dealing with diverse and extensive feature spaces



(4)

5)Gradient Boosting: Another ensemble learning method called gradient boosting generates a series of weak learners (usually decision trees) one after the other.To increase overall accuracy, it integrates the predictions of every weak learner and fixes mistakes produced by earlier models. Gradient Boosting minimizes a cost function by adjusting subsequent models based on the gradient of the loss concerning the model's prediction.Notable implementations include XGBoost, LightGBM, and AdaBoost, all of which contribute to high predictive accuracy.This method excels in capturing complex relationships within data and is well-suited for various machine learning tasks, including text classification.

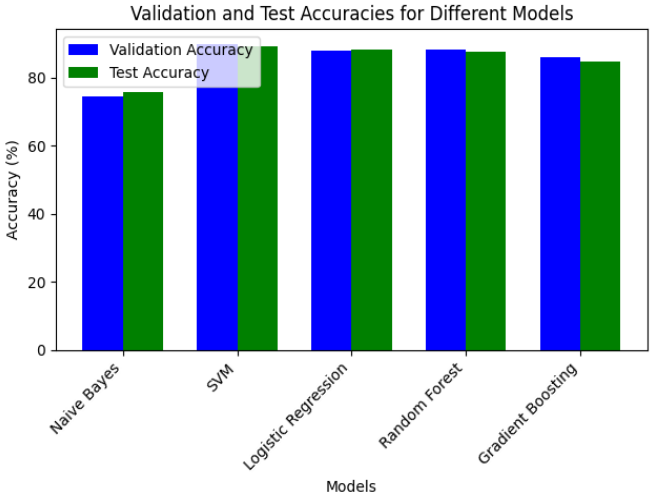


(5)

IV. RESULTS AND DISCUSSION

Four widely-used metrics are available for assessing the effectiveness of a text categorization system: accuracy, precision, recall, and F1 measure. These measures shed light on the classifier's efficacy and efficiency. The performance evaluation metrics that are employed are as follows:

*1)**Accuracy:*** The overall correctness of the classification findings is measured by accuracy. calculates the total number of cases that have been classified. The accuracy formula is:
Accuracy is calculated as (Number of examples accurately classified) / (Total number of occurrences).

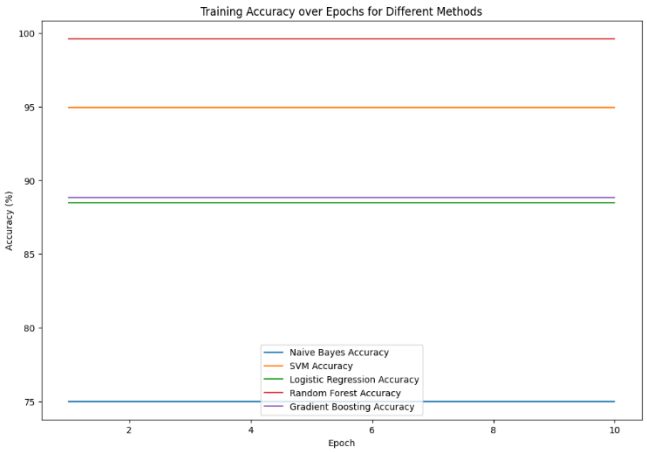


Our study highlights the importance of using appropriate performance metrics to evaluate text classification systems. Because it ignores false positives and false negatives, the accuracy statistic by itself might not give a clear view of the classifier's performance. The measurements for precision and recall offer light on how well the classifier can identify

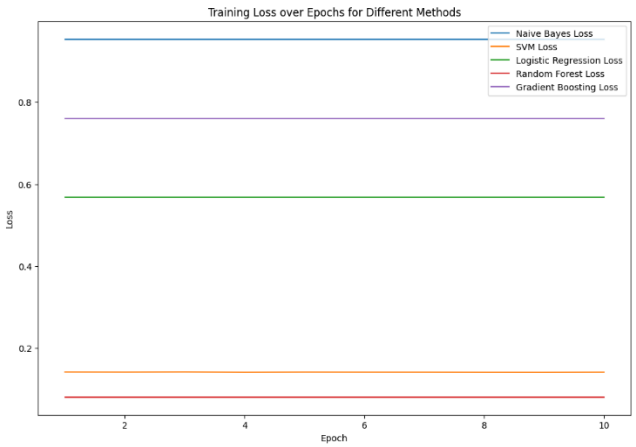
instances belonging to a specific category and avoid misclassification.

PERFORMANCE OF THE VARIOUS MODELS

Method	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
Naive Bayes	0.9709953335638639	74.4	0.809277699354689	0.5741437270721691
SVM	0.3162356256462033	89.64999999999999	0.8404835382067517	0.97602739506958
Logistic Regression	0.5652745460031929	88.0	0.760834937894994	0.8951348220405588
Random Forest	0.3719882190270494	88.25	0.5373837121460145	0.6580302796876167
Gradient Boosting	0.7949587602236797	85.85000000000001	0.9353104433606916	0.6215684906950822



(a)



(b)

It demonstrates the effectiveness of SVM-based text classification systems in classifying documents in Indian languages. We believe that our findings will be useful in developing more accurate and efficient text classification systems for Indian languages and improving access to information in non-English speaking populations.

CONCLUSION

In conclusion, text classification is an important text mining task, especially with the growing amount of large-scale data available online. The expansion of social media and the diverse languages used in India adds complexity to text classification. While there has been some progress in text classification for Indian languages, there is still much to explore in terms of text classification for Indian content.

Text categorization has demonstrated the effectiveness of supervised learning techniques including Naive Bayes, Support Vector Machines, Logistic Regression, Random Forest, and Gradient Boosting. Support vector machines and naive bayes have been especially useful in various linguistic applications.

However, it is important to note that supervised approaches depend on annotated training data, and moving the classifier to a new domain requires collecting annotated data specific to that domain. Unsupervised learning approaches have also been explored, which use methods like topic modeling, clustering, and lexical resources to find patterns or clusters inside the text data rather than depending on labeled data.

Overall, there is still a need for further research and exploration in text classification for Indian languages. More annotated data being available and the creation of language-specific methods and resources will contribute to improving the accuracy and performance of text classifiers for Indian languages.

ACKNOWLEDGMENT

One should not tally names found in the Notes and References. The IEEE LaTeX-style document used to prepare this model was created and maintained by Michael Shell and other collaborators, for whom Causal Productions is grateful. Look at the top of the IEEETran.cls file in the IEEE LaTeX release to view a list of contributors.

REFERENCES

[1] .P. M. Metev and V. P. Veiko, Laser-assisted microtechnology, 2nd ed., R . M. Osgood, Jr., editor. Berlin, Germany: Springer-Verlag, 1998.

[2] Sebastiani, F. (2002). Machine learning in automatic text classification. ACM Research in Computing, 34(1), 1-47.

[3] Nidhi and Gupta, V. (2012). Area wise classification of Punjab data. Proceedings of COLING 2012: Show Papers, 297-304.

[4]. Zheng, G. and Tian, Y. (2010). Suav Internet Text Classification System Model Raws li Naive Bayes International Conference on Electronic Products, Electronic Services and Electronic Entertainment (ICEEE), 1-4.

[5]. Murthy, K. N. (2003). Automatic classification of Telugu news articles. Department of Computer Thiab Information Science, University of Hyderabad.

[6]. Jayashree, R. (2011). Sentence level text analysis in Kannada. International Conference on Software Computing and Pattern Recognition (SoCPaR), 147-151.

[7]. Maliha, R.A. and Maliha, İ. (2009). Urdu text classification. Proceedings of the 7th International Conference on the Frontiers of Information Technology. [7]. Wapnik, V.N. (1995). The meaning of science education. New York: Springer.

[9] Rocchio, J. (1971). Relevant content in data collection. G. Salton (ed.), Intelligent Systems, 67-88.

[10] Ko, Y. and Seo, J. (2000). Automatic text classification via unsupervised learning. Proceedings of the 18th Communication Conference, 1, 453-459.

[11] Kaur, J. and Saini, J.R. (2014). Research and analysis of thought mining studies on Indo-Aryan, Dravidian and Tibeto-Burman languages. International Journal of Data Research and Emerging Technologies, 4(2), 53-60.

[12] Mansoor, M., Uzzaman, N. and Khan, M. (2006). N-Gram based text analysis of Bengali language in newspaper corpus. Proceedings of the 9th International Conference on Computer and Information Technologies