Text-Based Emotion Classifier

G. Dileep Chandu[a], G. Keerthi Vardhani[b], G. Vyshnavi[c]

[a] *20BCI7313, Computer Science & Engineering, Vellore Institute of Technology, Andhra Pradesh, India*
[b] *20BCE7620, Computer Science & Engineering, Vellore Institute of Technology, Andhra Pradesh, India*
[c] *20BCI7004, Computer Science & Engineering, Vellore Institute of Technology, Andhra Pradesh, India*

***Abstract*: In this digital age, a vast amount of documents in various Indian languages are available in digital form, posing the challenge of efficient retrieval and organization of these documents. Text classification, a field in text mining, offers a solution to this challenge by assigning classes to documents based on their content. This paper presents an analysis of text classification techniques applied to Indian language content, taking into account the unique challenges of natural language processing. The study reveals that supervised learning algorithms, such as Naive Bayes, Support Vector Machines, Artificial Neural Networks, and N-gram, have shown promising performance in text classification tasks. The paper highlights the significance of text classification in managing and organizing large volumes of textual data.**

***Keywords*: Classification, Naive Bayes, Natural Language Processing, Supervised Learning, Support Vector Machine.**

I. INTRODUCTION

The exponential growth of the World Wide Web has resulted in an immense accumulation of data, predominantly in the form of text. However, this abundance of information presents a challenge in terms of identifying relevant knowledge or information. Text classification addresses this challenge by categorizing a set of input documents into predefined classes, allowing for efficient organization and retrieval of information [1]. Text classification is a text mining technique that plays a crucial role in various applications, including document indexing, document organization, and hierarchical categorization of web pages. By automating the classification process, text classification offers significant advantages over manual classification methods, such as speed and efficiency.

Language serves as the primary medium for both written and spoken communication. With the utilization of Unicode encoding, text on the web can be found in diverse languages, introducing the complexities of natural language processing into text classification. Text classification, therefore, encompasses both text mining and natural language processing. The process involves combining information retrieval (IR) technology and machine learning (ML) technology to assign keywords to documents and classify them into specific categories. ML algorithms enable automatic categorization, while IR techniques represent text as features.In recent years, the growth of the internet in India has led to a surge in digital content creation 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 challenges posed by the diverse Indian language landscape, significant progress has been made in the development 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 aims to provide an overview of the various approaches employed in text classification in Indian languages and highlight the specific work carried out 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 examine the applications of text classification in Indian languages, including 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. Section II provides an overview of the steps involved in the text classification process. Section III discusses the various approaches employed in text classification and highlights the specific work carried out in the context of Indian languages.

II. LITERATURE SURVEY

In their work, Dongliang Xu et al. [1] proposed a microblog emotion classification model called CNN_Text_Word2vec,

which utilized a convolutional neural network (CNN) for feature extraction. The model achieved good classification results and outperformed other methods such as SVM, RNN, and LSTM in terms of emotional classification accuracy. However, one limitation of the model was the improper ranking between the extracted features.

Srishti Vashishtha et al. [2] developed a sentiment analysis system for social media posts using a set of fuzzy rules. Their approach involved multiple lexicons and datasets, integrating natural language processing (NLP) techniques and word sense disambiguation. The fuzzy system, based on a novel unsupervised nine fuzzy rule-based system, provided accurate sentiment values and addressed linguistic problems. The scheme outperformed other state-of-the-art methods, but it exhibited a high error rate, leading to inaccurate class fixation.

Jun Li et al. [3] introduced a multi-label maximum entropy (MME) model for user emotion classification in short texts. The MME model generated rich features based on multiple emotion labels and valence scores from users. The scheme successfully identified entities and provided relevant social emotions using generated lexicons. While the method was effective in classifying social emotions over sparse features, it had issues with overfitting.

Fazeel Abid et al. [4] developed a scheme that combined distributed word representations (DWRs) through a weighted mechanism on variants of recurrent neural networks (RNNs) and convolutional neural networks (CNNs) with weighted attentive pooling (WAP). The scheme addressed syntactic and semantic regularities, as well as out-of-vocabulary (OOV) words. The experimental analysis showed that the scheme achieved an accuracy rate of 89.67%. However, it had a limitation of inadequate feature extraction, leading to analysis errors.

Peng Wu et al. [5] proposed an Ortony-Clore-Collins (OCC) model and a convolutional neural network (CNN) based institutionalization method for sentiment analysis of Chinese microblogging systems. The scheme combined emotion cognition with deep learning and outperformed other state-of-the-art methods in terms of classification and recognition performance. However, the scheme had a complexity issue regarding microblog sentiment classification.

Muhammad Asif et al. [6] implemented sentiment analysis of multilingual textual data from social media to detect the intensity of extremist sentiments. The scheme effectively identified extreme sentiment from multilingual data and achieved an overall accuracy rate of 82%. The scheme outperformed existing techniques in terms of scalability and reliability. However, it had performance degradation in the recognition of multimodal sentiment. An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

III. PROPOSED WORK

A. Text Classification Process

The text classification process consists of several sub-phases, each playing a crucial role in achieving accurate classification results. Figure 1 illustrates the basic text classification process, which includes the following sub-parts: data collection, pre-processing, feature extraction, feature selection, building a classifier, and performance evaluation [1] [2]. The purpose and importance of each sub-phase are described below:

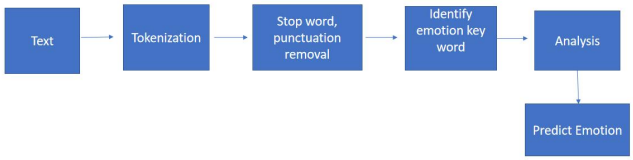


Fig. 1: Flowchart of proposed model

1) Data Collection

The first step in the classification process is building a corpus by collecting documents in various formats, such as .html, .pdf, .doc, and web content. These documents are used for training and testing the classifier

B. Pre-Processing

The pre-processing phase involves transforming the text documents into a clear word format. This step prepares the documents for further processing in text classification and involves the following common steps:

1) *Tokenization:*

Tokenization is a fundamental step in the pre-processing phase of text classification. It involves breaking down a text document into individual tokens or words, which are then used as features for classification. Tokenization is typically performed using natural language processing (NLP) techniques such as regular expressions, which can identify word boundaries and special characters. Additionally, tokenization can also take into account multi-word expressions or n-grams, which involve grouping together adjacent words to capture their combined meaning.

2) Removing stop words:

Stop words are common words that appear frequently in a language and do not provide much semantic meaning. Removing stop words can reduce the dimensionality of the feature space and help to improve the accuracy of the classification model. However, in some cases, removing stop

words may not be appropriate, especially if the stop words carry important information for classification. Therefore, it is important to carefully consider the relevance of stop words in each specific text classification task.

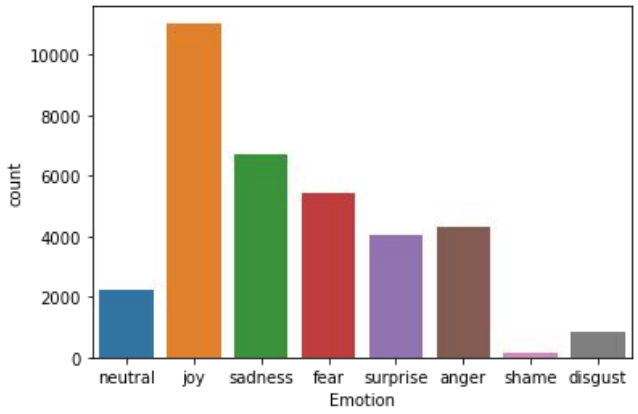


Fig. 2 :Data Set outcomes

3) 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.

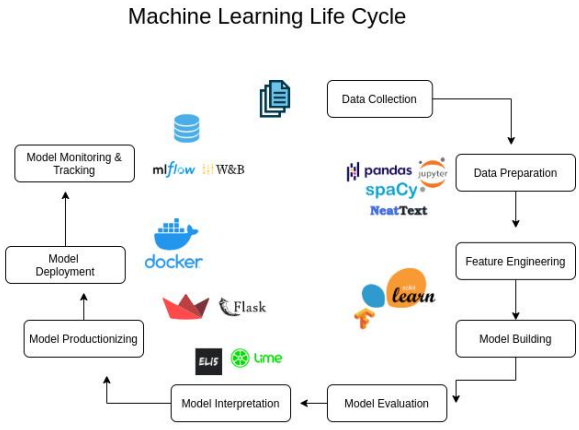


Fig. 3:Flowchart of methodology

C. Feature Extraction

Feature extraction is a pre-processing technique used to reduce the complexity of the documents and make them easier to handle. In this step, the documents are transformed from their full-text version to document vectors. The most commonly used document representation is the count vectorizer model (CVM), where documents are represented by vectors of words. However, CVM has limitations, including high dimensionality, loss of correlation between adjacent words, and loss of semantic relationships among terms. To address these issues, term weighting methods can be applied to assign appropriate weights to the terms

D. Feature Selection

After pre-processing and feature extraction, feature selection is a crucial step in text classification. It aims to construct a vector space that improves the scalability, efficiency, and accuracy of the text classifier. Feature selection involves selecting a subset of features from the original documents. This process is performed by identifying words with the highest scores according to a predetermined measure of word importance. The high dimensionality of the feature space is a major challenge in text classification, and various feature evaluation metrics are used, including information gain (IG), term frequency, Chi-square, expected cross-entropy, odds ratio, weight of evidence, mutual information, and Gini index [1].

E. Classification

The classification phase involves automatically categorizing documents into predefined categories. There are three main methods for document classification: unsupervised, supervised, and semi-supervised. In recent years, significant progress has been made in automatic text classification, particularly in machine learning approaches such as Bayes classifier, Logistic Regression, and support vector machines (SVMs).

1) Navie Bayes:

Naive Bayes is a simple probabilistic classifier that applies Bayes' theorem with strong independence assumptions. 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.

2) SVM:

SVM is a statistical classification method proposed by Vapnik. It seeks a decision surface to separate training data points into two classes and makes decisions based on the

support vectors. SVM is considered one of the best text classification methods and has been widely used in various studies.Support Vector Machines (SVM) have been extensively used in the field of text classification due to their ability to handle high-dimensional data with a small sample size. SVMs have proven to be effective in separating the data points into two classes by finding the hyperplane that maximizes the margin between the two classes.

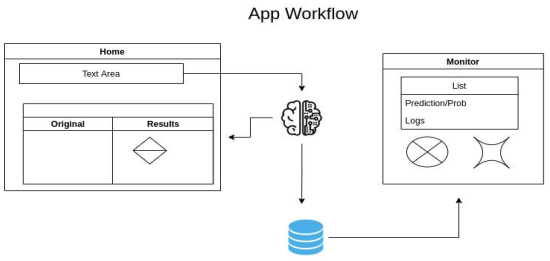
In text classification, SVM can be used to train a model to classify documents into different categories based on the presence or absence of specific keywords or features. The SVM model learns to identify the most important features that differentiate the classes and assigns a weight to each feature based on its importance. During the classification stage, the SVM model uses these weights to assign a document to the most appropriate category. SVM outperforms other classification algorithms in text classification tasks. SVM has been used for various text classification tasks such as sentiment analysis, topic modeling, and document classification. Additionally, SVM has been used in combination with other techniques such as feature selection and ensemble methods to further improve classification accuracy.

3) Logistic Regression:

A Logistic regression is a statistical method used to model the relationship between a binary dependent variable (i.e., a variable that can only take on two values, usually coded as 0 and 1) and one or more independent variables. It is a type of regression analysis that is commonly used in machine learning and predictive modeling.The goal of logistic regression is to estimate the probability that a given observation belongs to a certain class, based on the values of the independent variables. The output of the logistic regression model is a logistic function, also known as a sigmoid function, which maps any real-valued input to a value between 0 and 1.

The logistic function takes the form:
$p(x) = 1 / (1 + e^{(-z)})$

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



IV. RESULTS AND DISCUSSION

The performance of a text classification system can be evaluated using four commonly used metrics: accuracy, precision, recall, and F1 measure. These metrics provide insights into the effectiveness and efficiency of the classifier. The following metrics are used for performance evaluation

1) *Accuracy:*

Accuracy measures the overall correctness of the classification results. It calculates the ratio of correctly classified instances to the total number of instances. The formula for accuracy is:
Accuracy = (Number of correctly classified instances) / (Total number of instances)

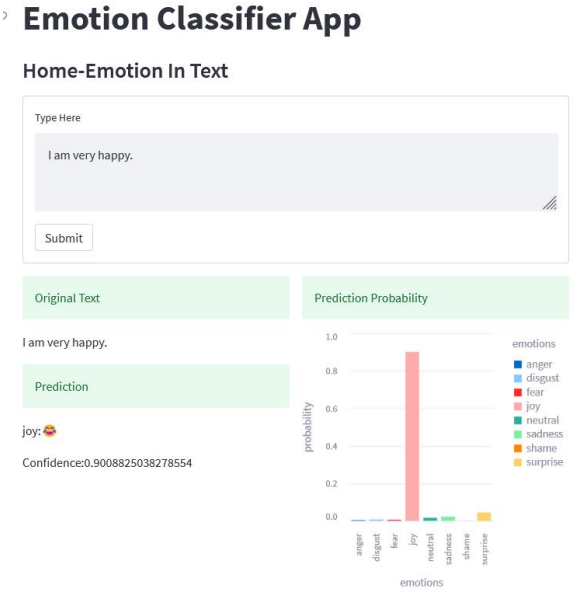


Fig. 4:Testing the app

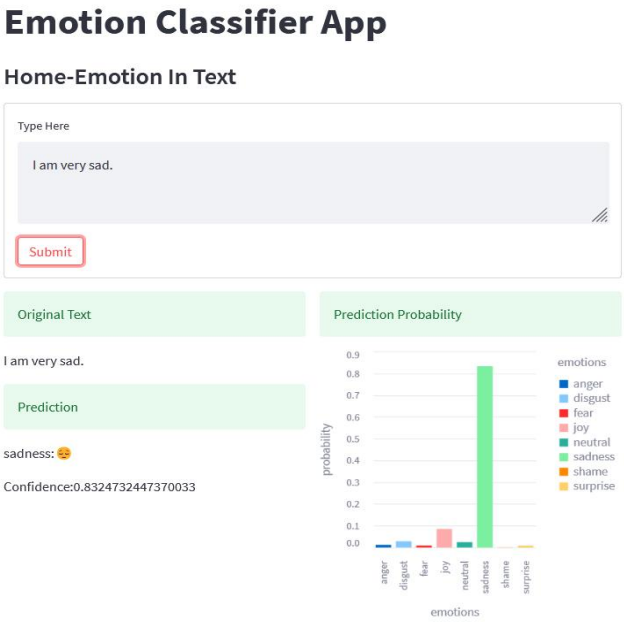


Table :1 Results of SVM, NB, LR algorithms

Algorithm	**Accuracy**
SVM	82%
Navie Bayes	85%
Logistic Regression	90%

Our study highlights the importance of using appropriate performance metrics to evaluate text classification systems. The accuracy metric alone may not provide a complete picture of the classifier's effectiveness, as it does not take into account false positives and false negatives. The precision and recall metrics provide insights into the classifier's ability to correctly identify instances belonging to a specific category and avoid mis-classification.

It demonstrates the effectiveness of Logistic Regression-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.

IV. CONCLUSION

In conclusion, text classification is a crucial task in the field of text mining, particularly with the increasing availability of large amounts of data on the web. 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.

Supervised learning approaches, such as Naive Bayes, Support Vector Machines, Logistic Regression have shown success in text classification. Naive Bayes and Support Vector Machines have been particularly effective in different language contexts.

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 do not rely on labeled data but instead aim to discover patterns or clusters within the text data using techniques such as lexical resources, clustering, and topic modeling.

Overall, there is still a need for further research and exploration in text classification for Indian languages. The availability of more annotated data and the development of language-specific techniques and resources will contribute to improving the accuracy and performance of text classifiers for Indian languages.

V. ACKNOWLEDGMENT

The heading of the Acknowledgment section and the References section must not be numbered.

Causal Productions wishes to acknowledge Michael Shell and other contributors for developing and maintaining the IEEE LaTeX style files which have been used in the preparation of this template. To see the list of contributors, please refer to the top of file IEEETran.cls in the IEEE LaTeX distribution.

VI. REFERENCES

[1] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[2] . Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Surveys, 34(1), 1-47.

[3] 2. Nidhi, & Gupta, V. (2012). Domain-Based Classification Punjabi Text Documents. Proceedings of COLING 2012: Demonstration Papers, 297-304.

[4] 3. Zheng, G., & Tian, Y. (2010). Chinese web text classification system model based on Naive Bayes. International Conference on E-Product E-Service and E-Entertainment (ICEEE), 1-4.

[5] 4. Murthy, K. N. (2003). Automatic Categorization of Telugu News Articles. Department of Computer and Information Sciences, University of Hyderabad.

[6] 5. Jayashree, R. (2011). An analysis of sentence level text classification for the Kannada language. International Conference of Soft Computing and Pattern Recognition (SoCPaR), 147-151.

[7] 6. Ali, R. A., & Maliha, I. (2009). Urdu Text Classification. Proceedings of the 7th International Conference on Frontiers of Information Technology.

[8] 7. Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. New York: Springer.

[9] 8. Rocchio, J. (1971). Relevance Feedback in Information Retrieval. In G. Salton (Ed.), The SMART System, 67-88.

[10] 9. Ko, Y., & Seo, J. (2000). Automatic Text Categorization by Unsupervised Learning. Proceedings of the 18th conference on Computational linguistics, 1, 453-459.