EasyChair Preprint

№ 15739

Classification of Hindi News Articles Using Machine Learning Models with Challenges and Solutions

Subhashini Spurjeon Kashi and Ani Thomas

January 20, 2025

Classification of Hindi News Articles Using Machine Learning Models with Challenges and Solutions

Subhashini Spurjeon Kashi[1], Dr. Ani Thomas[2], Department of Computer Science and Engineering, Bhilai Institute of Technology, Durg (Chhattisgarh), India
k.subhashini@bitdurg.ac.in, ani.thomas@bitdurg.ac.in

Abstract:

In today's modern digitized world, large amount of Hindi text documents are generated and shared through many sectors, including public organizations, news portals, government webpages, and commercial sectors. These news documents need to be classified into distinct classes such as business, health, science, politics, and sports. Text classification is essential due to the overwhelming amount of unorganized data that exists. Hindi news agencies still rely on manual sorting due to the lack of a dedicated Hindi text classifier. While English text classification is well-established and has ample resources, Indian languages, particularly Hindi, lack standardized benchmarks. Hindi, one of the most popular and used languages in the world, faces challenges in text processing. Despite the progress made in text summarization, keyword extraction, and information retrieval, the creation of classifiers for dividing Hindi news articles into predefined categories is still lacking in several areas. This paper addresses this gap by preprocessing a collection of standard Hindi news articles at various levels—word, sentence, paragraph, and document. The paper also explores feature extraction techniques and applies machine learning classifiers to categorize the articles. Classifying Hindi news articles presents unique difficulties due to the language's intricate letter combinations, conjuncts, sentence structures, and multi-sense words.

Keywords:

Newswire articles, Preprocessing, Text classifier, NLP, keyword extraction, Machine Learning

1. Introduction

Numerous effective systems are present for classifying text documents using natural language processing features, supervised learning, and various classifiers have been suggested by the widespread availability of extensive text documents and the need to create clustered documents of the same type. However, most research contributions have been made in English text classification systems[1]. Such systems encountered various difficulties related to document misclassification, ambiguity removal, curse of dimensionality, dimensionality reduction, loss of important information, etc. Many academics and researchers are currently involved in mono/bi/tri/multilingual text processing and its relevance in distributed and real-time situations, looking at linguistic-based viewpoints[2]. Hence, this research paper uses NLP and Machine Learning models to build and put forward a new front for text classification systems for Hindi newswire articles. The article also investigates the various preprocessing methods used in the current research project and assesses how they affect text classification using NLP and machine learning algorithms. The main issue with using the Hindi text classification system is that, in most of the cases, the meaning or context of a sentence might get lost in the transition during the classification. Hence, a method still needs to be developed for Devanagari text classification[3]. This is the main reason that editors still classify news articles manually in India and hence need a system that can automatically classify them in the proper class based on their context[4]. For Classification of Hindi Newswire article, Classification system needs some predefined classes like: देश, विदेश, व्यापार, खेल, मनोरंजन, विविध, स्वास्थ्य, मौसम as shown in Figure 1.

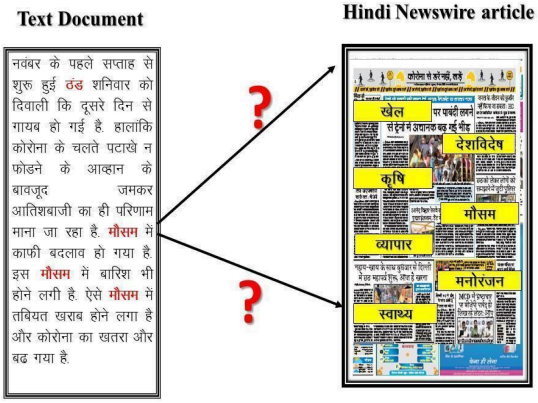


Figure. 1 Classification system with some predefined classes

NLP and Machine Learning techniques can efficiently organize Hindi Newswire articles into various sections. Since NLP and machine learning algorithms cannot work directly on the extracted text, there is a need for pre-processing and feature engineering techniques. Such pre-processing is a challenging task due to the presence of multisense words, a clearly defined, organized set of vocabulary and grammatical rules, and a vast collection of conjunctions and their several arrangements that make it different and unique from other methods while requiring an advanced crucial processing. The main drawback of working with Hindi language datasets is the need for pre-existing NLP libraries for the Hindi scripted data. Other issues in working with Hindi data are keyword and key-phrase extraction, Idiom recognition, word sense disambiguation, named entity recognition, identification and correction of word spellings, and sentence parsing and linguistic-based techniques [2].

This article focuses on overcoming these drawbacks and creating a Hindi Text Classification Model for newswire articles with maximum accuracy. The ability to categorize text documents is highly remarkable these days. One of the essential fields in text classification is News classification. This research paper is based on predominantly classifying Hindi newswire articles.

Newswire articles offer insights on several news topics catering to a vast audience. They can be among many categories like politics, sports, Science, and Technology. Due to the lack of classification systems, newspaper organizations manually classify articles and will need more classification systems to reduce the burden of manual sorting, even in the 21st century. Hence, Integrating Hindi news categorization models into an online news portal can eliminate the tedious process of manually classifying each piece into a category.

2. Literature Review

This section discusses work done in Hindi text classification and preprocessing text documents and provides a comprehensive summary of related research work.

Extraction of Events and Classification in Hindi: Event extraction is an essential task that is very widely performed in the English language. They investigated both event trigger detection and argument detection as sequence labeling problems. They have applied CNN, BiLSTM+GRU+CNN, and BiLSTM+GRU+self attention networks to extract text features automatically. BiLSTM+GRU+CNN is performing better than other models. Detection of Multi-word triggers is a crucial task and needs special consideration. With increased distance, linking an event trigger with its associated argument trigger becomes more challenging. They intend to look into Event Realis status classification and Event coreference resolution in the future[1]. Event extraction is a challenging task where an event can be detected along with relevant information like time, place, agent, intensity, etc[3]. Event trigger depends on Event trigger and event argument and comes under the area of neural relation extraction. The system comprises BiLSTM, CNN, and MLP layers, showing promising results. The system performs well at 60% of its total test instances. The system is unable to establish a connection between the event and the argument when the argument has some numeric value Ex: 500 लोग घायल हुए. The system needs more annotated data[5]. The system must perform event argument linking all over the text document[3]. The system needs clarification when long phrases with similar meanings describe an event. Improper balancing in the distribution of event class. Due to the data sparsity problem, there is a need to develop a stacked-based classification with Bi-LSTM as a base model followed by the CRF model. It is not possible to provide the link between event-argument[4]. Recent results in text classification using some deep learning techniques, such as CNN, RNN, LSTM, and BiLSTM, have been significant[6].

Hindi Text Classification using ML: Training Processor and Testing Processor are the two phases of the Text Classification module using SVM. The subject-object-verb (SOV) structure is used in Hindi sentences[2]. Poems in Hindi express various feelings that are distinct from one another. Naive Bayes and SVM classifiers were used to extract and select ras features. The modal will classify Hindi poems based on ras, and they have used emotional and sentimental features[7]. Proverb Identification and Keyword matching are two methods that will create headings or titles for Hindi stories. The NLP approach is suitable for creating a title for a specific story. The story can be predicted from the title[8]. Data sets of generic Hindi news titles with labels such as negative, positive, and natural remarks have been used to extract text features. They have used NB, Logistic, Random Forest, and SVM to extract and categorize sentiment features[9]. A machine learning method that creates an ensemble from different classifiers was recently developed to detect the polarity of sentiment in Hindi and Bengali tweets. Various classifier combination methods have been used to incorporate Nave Bayes multinomial with character n-gram features, Nave Bayes multinomial with word n-gram features, and SVM with unigram features into an ensemble[10]. Recent results in the classification of text using some machine learning algorithms, such as SVM, NB, LR, RF, and many more, have been significant[11][12][13][14].

Analyzing and classifying Hindi text is challenging, and more experimentation needs to be done. Using machine and deep learning directly on the dataset is complicated, so we must preprocess data before applying it to the system. The pre-processing includes cleaning up data, stopwords removal, tokenizing, and stemming. Then, logistic regression, Naïve Bayes with TF-IDF, LSTM, and many other approaches can be applied and compared for the appropriate results[15][16]. Preprocessing methods play a vital role in text mining techniques and

applications. It is the first and most crucial step in the text-mining process. The authors discussed critical preprocessing efforts, stopword removal, stemming, TF/IDF algorithms, and many more algorithms[17][18][19][20].

3. Methods and materials

In this section, the methods implemented to carry out the various activities in this dataset cleaning and preprocessing stage were discussed, and we further propose a model for the classification of Hindi newswire articles.

3.1 Dataset Collection: A dataset is a collection of many separate raw pieces of data but can be used to train an algorithm to find predictable patterns inside the whole data set. Data is one of the most essential components of NLP and machine learning. One of the biggest problems in Hindi text processing is the unavailability Of Hindi Data Set. The system needs Hindi News Articles Belonging to Different Categories for the Classification Model. So these news articles were collected from the BBC Hindi News Dataset on Kaggle and GitHub. Some data has also been taken from the Disaster dataset from IIT Patna. Usually, a dataset is used for more than just training purposes. A single training data set that has already been processed is generally split into several parts, which is needed to check how well the model's training went. For this purpose, a testing data set is usually separated from the data.

3.2 Pre-processing of Hindi Text: Preparing a dataset for a machine learning classifier model is a very important technique used called data pre-processing. It is the initial and the most crucial stage in developing a classification model. Actual data has noise, missing values, and occasionally in an undesirable form, and machine learning models cannot be applied directly to it. Pre-processing of raw data is a necessary step in cleaning up the data and preparing it for a machine learning model, improving its accuracy and effectiveness.

The following steps were followed to prepare the dataset for the application of the classification model:

i) Dataset Cleaning: Cleaning is the process of removing unwanted and noisy data. It also focuses on converting the dataset into a unified format.

The following steps were taken to clean the data:
1) The first step in dataset cleaning was removing empty, redundant, unwanted advertisement lines. The following Figure 2 shows the Hindi news article before cleaning. Lines 4-9 are blank lines, and the line highlighted in blue color indicates the advertisement line.

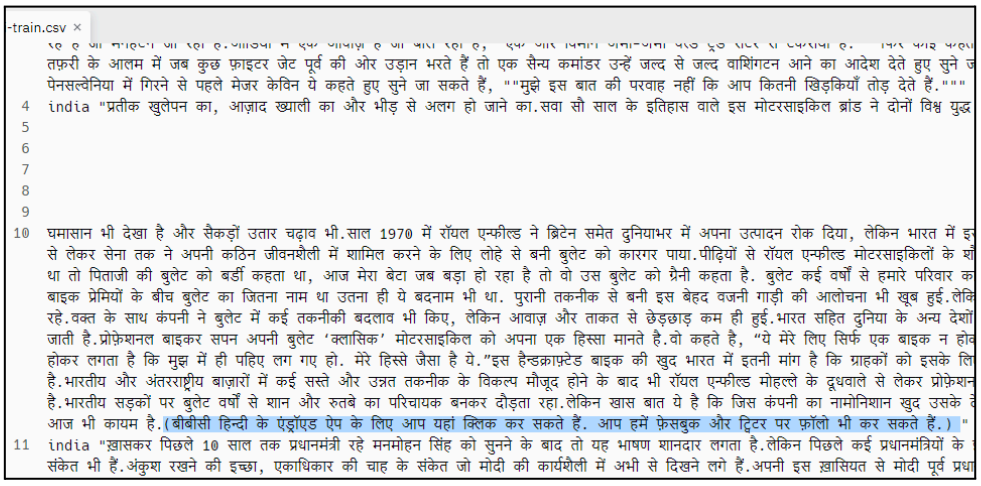


Figure 2: Dataset Before removing blank spaces and unwanted lines

After the cleaning process, the dataset looks like what is shown in the following screenshot. The blank lines are now deleted, and no article contains the advertisement lines.

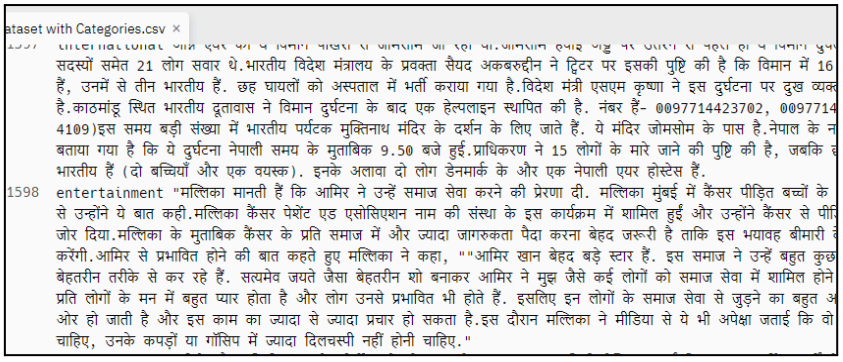


Figure 3: Dataset After removing blank spaces and unwanted lines

2) The next step is to rectify the spacing in the articles. As shown in Figure 4 below, the article in line 11 got split into lines 11 to 16.

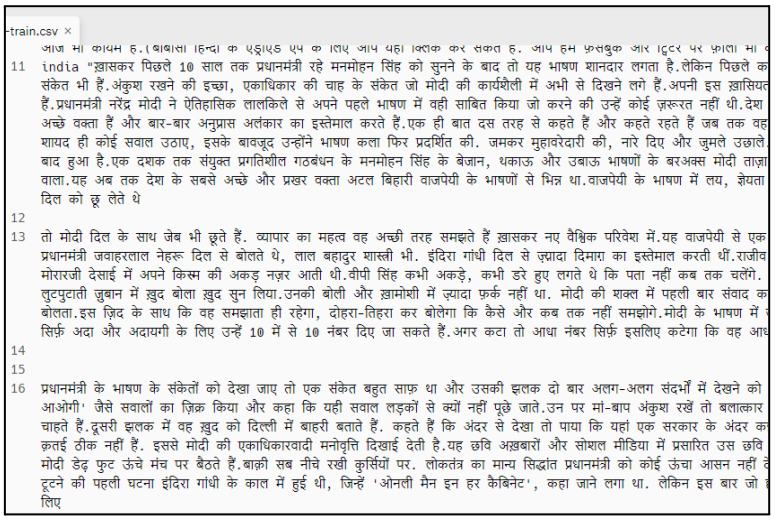


Figure 4: Dataset Before rectifying spaces

Such multi-row articles were rectified. Now, the dataset contains only one article in one row, as shown below.

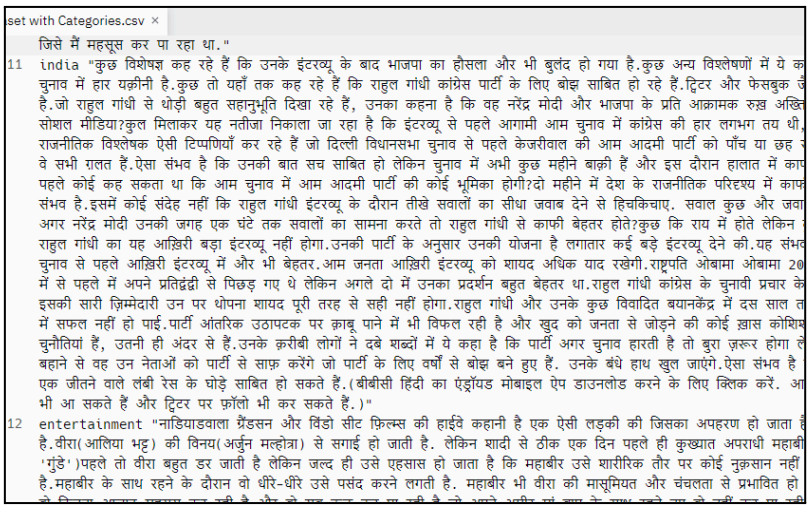


Figure 5: Dataset After rectifying spaces

3) Next step is to enclose each article within double quotes. Figure 6 shows that the article in line 2 is not enclosed within double quotes, whereas the article in line 3 is enclosed.

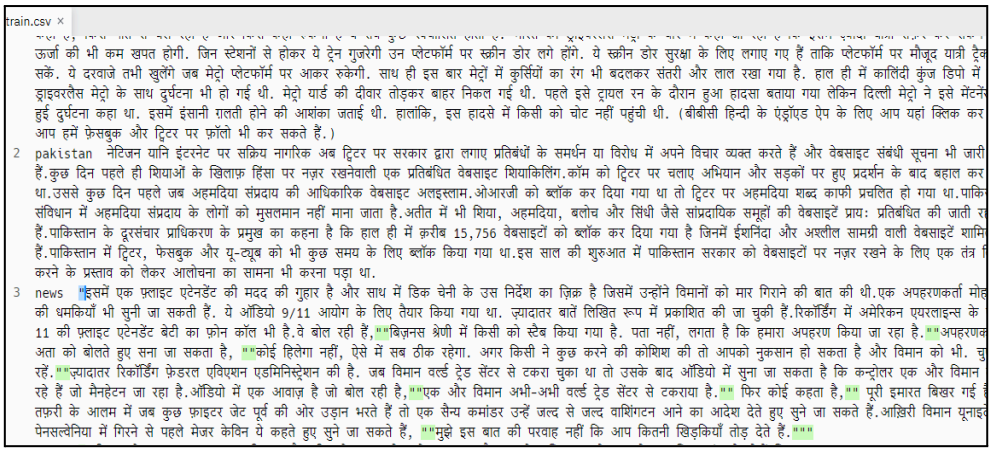


Figure 6: Dataset Before applying double quotes

All the articles were processed and enclosed within double quotes.

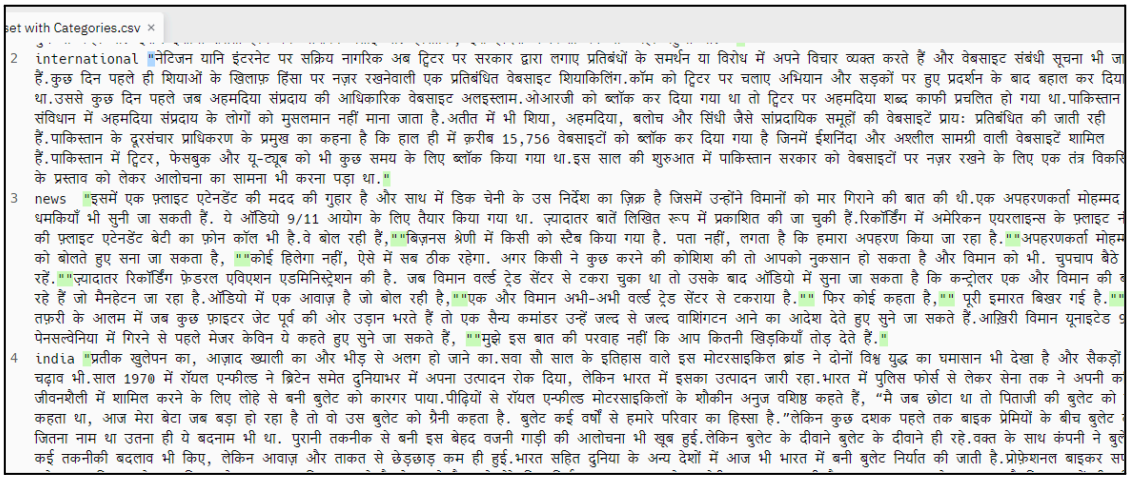


Figure 7: Dataset After applying double quotes

Categorization: The following Table 3 shows the categories and subcategories created in the dataset for the classification process. Figure 8 shows the categories and subcategories converted to Hindi Language. Then, the articles are manually tagged in each data row according to the type of categories and sub-categories. Each article unit is categorized into three columns, as shown in figure 9.

Main Category	Sub - Category	Location
India	Accident	State or City
International	Business	Country
	Crime	
	Disaster	
	Education	
	Entertainment	
	General	
	Healthcare	
	Political	
	Science and Technology	
	Social Media	
	Sports	
	War/Protest	

Table 1: Categorization Summary

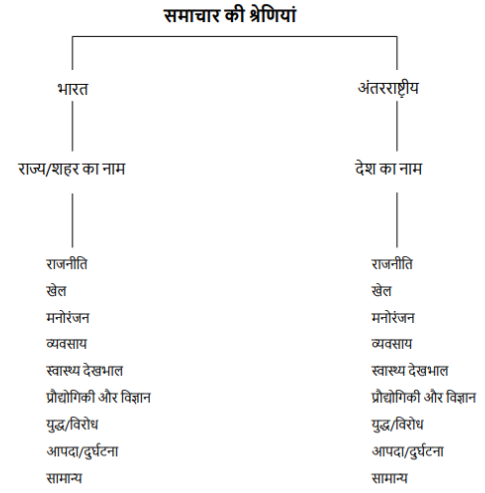


Figure 8: Categorization for tagging of articles

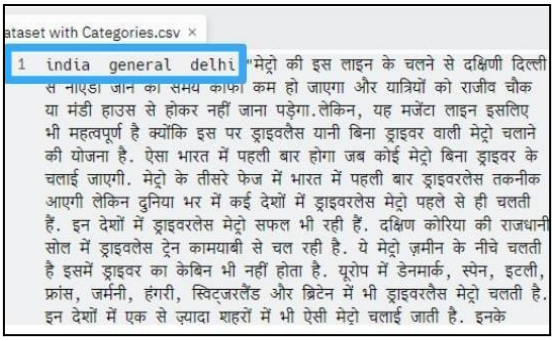


Figure 9: Article with categories, sub-categories, and location

ii) Tokenizing: Tokenization involves dividing the large text document into smaller units. It splits the original large text into small tokens, such as words and sentences. These tokens aid in context analysis or model preparation for language processing[21]. By examining the sequence of the words in the text, tokenization aids in understanding the meaning of the text. Ex: sample text "**मैं अपने स्तर पर सर्वश्रेष्ठ प्रयास करूंगा**" can be tokenized into

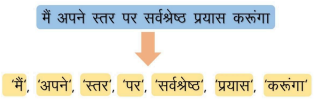
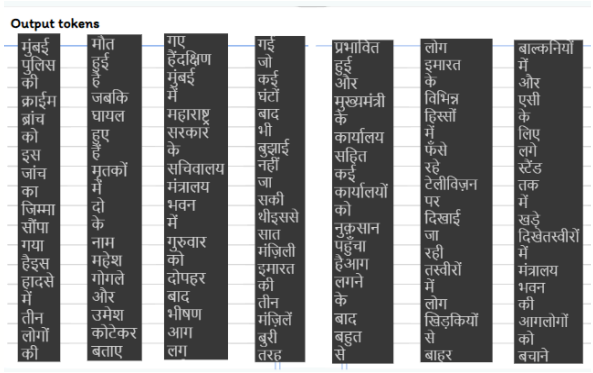


Figure 10: Tokenization Result

The first step is to tokenize the Hindi sentences into Hindi words. Let us consider the below given sentence as an input to our Tokenizer function, then the sentence count, the number of tokens, and the length of the text is given as follows:

INPUT: "मुंबई पुलिस की क्राईम ब्रांच को इस जांच का जिम्मा सौंपा गया है.इस हादसे में तीन लोगों की मौत हुई है जबकि **10** घायल हुए हैं. मृतकों में दो के नाम महेश गोगले और उमेश कोटेकर बताए गए हैं.दक्षिण मुंबई में महाराष्ट्र सरकार के सचिवालय मंत्रालय भवन में गुरुवार को दोपहर बाद भीषण आग लग गई जो कई घंटों बाद भी बुझाई नहीं जा सकी थी.इससे सात मंज़िली इमारत की तीन मंज़िलें बुरी तरह प्रभावित हुई और मुख्यमंत्री के कार्यालय सहित कई कार्यालयों को नुक़सान पहुँचा है.आग लगने के बाद बहुत से लोग इमारत के विभिन्न हिस्सों में फँसे रहे. टेलीविज़न पर दिखाई जा रही तस्वीरों में लोग खिड़कियों से बाहर बाल्कनियों में और एसी के लिए लगे स्टैंड तक में खड़े दिखे.तस्वीरों में: मंत्रालय भवन की आगलोगों को बचाने के लिए अग्निशमन विभाग के क्रेनों और हेलिकॉप्टर तक की मदद लेनी पड़ी.कम से कम **11** लोग घायल हुए हैं. इनमे से कुछ की हालत गंभीर है."

OUTPUT:



Some difficulties and issues to be solved: It needs to tokenize the numerals. It also combines the words before and after the full stop, as shown in the following text.

iii) Stopword removal: Many words in natural languages are applied as syntactic units to finish sentences and contain little information. Additionally, it should be noted that, compared to short texts, the effect of stopword removal on retrieval performance is maximum on large documents. Stopword elimination enhances text retrieval performance and other computational tasks like text classification, sentiment classification, text summarization, and machine translation[22][23].

iv) Parts of Speech Tagging: It is the most essential stage in developing NLP Applications and related fields. POS is a grammatical arrangement that assigns tags to every word in the text. It is the logic behind the choice

made by researchers to describe this task as a sequence-labeling task, where all words are interpreted as sequences that require labeling. Each tag for the word is recognized in a context by the preceding word or its combination of tags. POS tagging is applied in many application areas, such as parsing, where observation and word tags are converted into chunks that can be integrated to produce the complete parse tree for a text document. Hindi is similar to English but does not have precise equivalents for parts of speech like prepositions, nouns, pronouns, verbs, adverbs, adjective interjections, and conjunctions. In Hindi, the POS of a word or lexis depends on its morphological characteristics and where it appears in a sentence[24][25][26]. Take a look at one of the text samples from Figure 11. In the two statements, the identical word, "सोना" is given a different label. As it refers to an item (Gold Ornament) in the first instance, it is classified as a common noun. In the second instance, it is referred to as a verb because it relates to the speaker's experience (or feelings). Looking at the word or tag combinations of the nearby comments about the ambiguous word—the word with several tags—can address this issue.

Numerous studies on POS tagging have been conducted over the years. All of the attempts can be broadly divided into three categories. They are rule-based approaches, where a linguistic expert must create rules for categorizing words. Statistical techniques, where categorizing words using mathematical formulas. Hybrid systems, which combine rule-based and statistical methods. In European languages, POS taggers are typically created using a machine-learning technique. However, in Indian languages, we still need a straightforward, effective process.

There are 8 essential POS: Nouns (naming word), Pronouns (replace a noun), Adjectives (describing word), Verbs (action word), Adverbs (describe a verb), Preposition (showing relationship), Conjunction (joining word) and Interjections (expressive word). Mostly, it is partitioned into sub-sections. Noun is divided into proper nouns, common nouns, concrete nouns, etc.

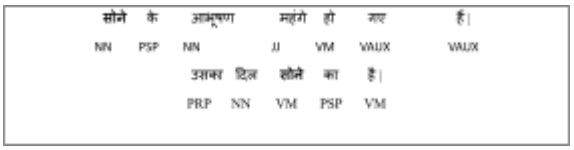


Figure 11: Parts of Speech Tagging Example

Ex: Sample text = "इराक के विदेश मंत्री ने अमरीका के उस प्रस्ताव का मजाक उड़ाया है , जिसमें अमरीका ने संयुक्त राष्ट्र के प्रतिबंधों को इराकी नागरिकों के लिए कम हानिकारक बनाने के लिए कहा है ।" The tagged text then will be as follows,

tagged_words = (tnt_pos_tagger.tag(nltk.word_tokenize(text))) print(tagged_words) [OUTPUT]: [('इराक', 'NNP'), ('के', 'PREP'), ('विदेश', 'NNC'), ('मंत्री', 'NN'), ('ने', 'PREP'), ('अमरीका', 'NNP'), ('के', 'PREP'), ('उस', 'PRP'), ('प्रस्ताव', 'NN'), ('का', 'PREP'), ('मजाक', 'NVB'), ('उड़ाया', 'VFM'), ('है', 'VAUX'), (',', 'PUNC'), ('जिसमें', 'PRP'), ('अमरीका', 'NNP'), ('ने', 'PREP'), ('संयुक्त', 'NNC'), ('राष्ट्र', 'NN'), ('के', 'PREP'), ('प्रतिबंधों', 'NN'), ('को', 'PREP'), ('इराकी', 'JJ'), ('नागरिकों', 'NN'), ('के', 'PREP'), ('लिए', 'PREP'), ('कम', 'INTF'), ('हानिकारक', 'JJ'), ('बनाने', 'VNN'), ('के', 'PREP'), ('लिए', 'PREP'), ('कहा', 'VFM'), ('है', 'VAUX'), ('।', 'PUNC')] Now this tagged data can be used to train the model

The main issue here is that every word from the text is not tagged correctly, because of which the tag "Unk" is assigned most of the time. This problem can be overcome by stemming the word using probability with frequency for the next term, handling compound words, and adding more tagged sentences to NLTK using Google Translate to translate and get the tag. In this model, we have used the Google Translator API to solve the problem of "Unk" tags and then appending them to the NLTK Indian Corpus, which gives good results.

By evaluating the forward and backward probabilities of tags along with the sequence given as input, a POS tagger based on HMM assigns the best tag to a word. The following equation explains these phenomena.[27].

$$P(xi/yi) = P(xi/xi\text{-}1) \,.\, P(xi+1/xi) \,.\, P(yi/xi) \qquad (1)$$

Here, P(xi/xi-1) is the probability of a current tag given the previous tag, and P(xi+1/xi) is the probability of the future tag given the current tag. This captures the transition between the tags.

$$P(xi/xi\text{-}1) = freq(xi\text{-}1, xi) \,/\, freq(xi\text{-}1) \qquad (2)$$

These probabilities are computed using Equation 2. The frequency count of the two tags are observed together in the corpus divided by the frequency count of the immediately preceding tag followed independently in the corpus yields the likelihood of each tag transition. This will happen because it is known that some tags are more likely to appear before other tags. For instance, a common noun (NN) will come after an adjective (JJ), not a

postposition (PSP) or a pronoun (PRP) shown in the figure. The figure displays this case. POS tags for Hindi are as follows:

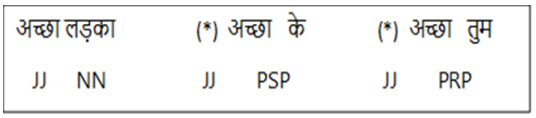


Figure 12: Parts of Speech Tagging Example

S.No.	Tag	Description (Tag used for)	Example
1	NN	Common Nouns	लड़का लड़के किताब पुस्तक
2	NST	Noun Denoting Spatial and Temporal Expressions	ऊपर पहले बाहर आगे
3	NNP	Proper Nouns	मोहन राम सुरेश
4	PRP	Pronoun	वह उसे तुम
5	DEM	Demonstrative	वह उस
6	VM	Verb Main	खाता सोता रोता खाते सोते रोते
7	VAUX	Verb Auxillary	हैं हुए कर
8	JJ	Adjective	सांस्कृतिक पुराना दुपहिया
9	RB	Adverb	जल्दी धीरे धीमे
10	PSP	Postposition	में को ने
11	RP	Particles	भी तो ही
12	QF	Quantifiers	बहुत थोड़ा कम

Table 2: PoS Tags for Hindi[27]

POS Tagging: Let us consider the below given Hindi news dataset as an input, our PoS Tagger will provide grammatical tags to each and every word in the sentence.

INPUT: मेट्रो की इस लाइन के चलने से दक्षिणी दिल्ली से नोएडा जाने का समय काफी कम हो जाएगा और यात्रियों को राजीव चौक या मंडी हाउस से होकर नहीं जाना पड़ेगा.लेकिन, यह मजेंटा लाइन इसलिए भी महत्वपूर्ण है क्योंकि इस पर ड्राइवलैस यानी बिना ड्राइवर वाली मेट्रो चलाने की योजना है.
OUTPUT:
==========New Tagged words==========
('मेट्रो', 'NNP'), ('की', 'PREP'), ('इस', 'PRP'), ('लाइन', 'NN'), ('के', 'PREP'), ('चलने', 'VBG'), ('से', 'PREP'), ('दक्षिणी', 'JJ'), ('दिल्ली', 'NNP'), ('से', 'PREP'), ('नोएडा', 'NN'), ('जाने', 'VNN'), ('का', 'PREP'), ('समय', 'NN'), ('काफी', 'INTF'), ('कम', 'JVB'), ('हो', 'VFM'), ('जाएगा', 'VAUX'), ('और', 'CC'), ('यात्रियों', 'NNS'), ('को', 'PREP'), ('राजीव', 'NNPC'), ('चौक', 'NN'), ('या', 'CC'), ('मंडी', 'NN'), ('हाउस', 'NNP'), ('से', 'PREP'), ('होकर', 'VRB'), ('नहीं', 'NEG'), ('जाना', 'VAUX'), ('पड़ेगा.लेकिन', 'NN'), (',', 'PUNC'), ('यह', 'PRP'), ('मजेंटा', 'NN'), ('लाइन', 'NN'), ('इसलिए', 'RB'), ('भी', 'RP'), ('महत्वपूर्ण', 'JJ'), ('है', 'VFM'), ('क्योंकि', 'IN'), ('इस', 'PRP'), ('पर', 'PREP'), ('ड्राइवलैस', 'NN'), ('यानी', 'CC'), ('बिना', 'NEG'), ('ड्राइवर', 'NN'), ('वाली', 'PREP'), ('मेट्रो', 'NNP'), ('चलाने', 'VNN'), ('की', 'PREP'), ('योजना', 'NN'), ('है', 'VFM').

v) Keyword Extraction: To extract meaningful information from text documents, keyword extraction is frequently used. It is an efficient method of extracting the most appropriate words and phrases from input text. This method automatically pulls a document's most important words and expressions [27]. After parts of speech tagging, we can apply the keyword extraction function on the text to identify trigger words for every category. Let us consider the below Hindi sentence as an input to the keyword extraction function, then the output obtained is given as follows:

INPUT: "इराक के विदेश मंत्री ने अमरीका के उस प्रस्ताव का मजाक उड़ाया है , जिसमें अमरीका ने संयुक्त राष्ट्र के प्रतिबंधों को इराकी नागरिकों के लिए कम हानिकारक बनाने के लिए कहा है ।"
OUTPUT: {'अमरीका', 'इराक', 'विदेश मंत्री', 'प्रस्ताव', 'नागरिकों', 'प्रतिबंधों', 'अमरीका संयुक्त राष्ट्र'}

3.3 Proposed Approach

In this section, the system followed the classification approach for classifying Hindi Newswire articles into predefined classes. Figure 13 block diagram depicts the overall classification system. The complete system is broadly divided into two modules. The first module includes the dataset preparation and preprocessing, and the second is a thorough machine-learning model preparation.

The preprocessing techniques mentioned in this paper will be used alongside machine and deep learning algorithms to form a Text Classification Model for Newswire Articles. The machine learning techniques used for classification are NB, logistic, SVM, and deep learning techniques are CNN, BiLSTM, and BERT Model. The following block diagram shows the various activities performed at each stage in the proposed Hindi News Classifier model.

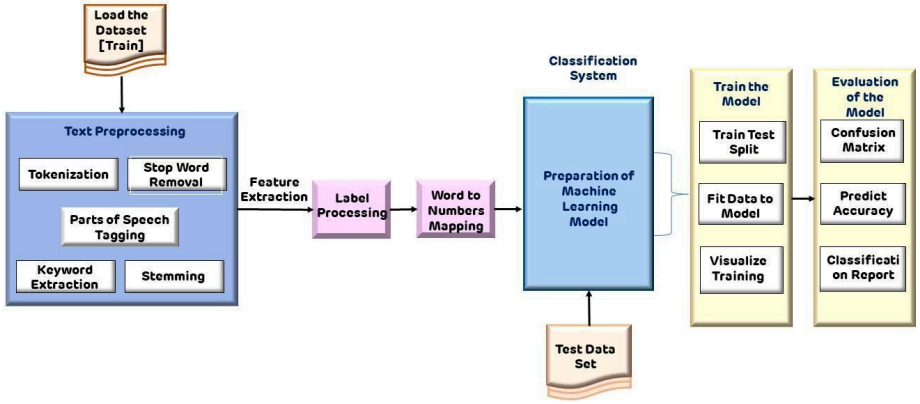


Figure 13: Block Diagram for Classification of Hindi Newswire articles

4. RESULTS AND DISCUSSION

This section contains the proper outcomes after applying the preprocessing methodologies to the classification system.

Label Processing: Label processing is an essential step in the categorization process that takes raw data and converts it into a format that can be analyzed and understood by computers and machine learning. In this step, labels for all classes are generated and put into a single list such that the labels are uniform throughout. Labels are created according to each class length. We have divided the dataset into news_articles and Labels. We have two levels of classification: one is binary classification, and another is multi-class classification.
The Level-1 labels for the binary classification is as follows: India -1, International - 0
The Level-2 labels for the multi class classification is as follows: 0-12 labels for 13 different news categories

Word to Number Mapping: Word embedding, commonly referred to as word vector, is a method for representing both documents and words. A numeric vector input allows words with similar meanings to have the exact representation. It can also approximate the purpose and define a word in a lower dimensional space. We have converted the news article and labels array into one hot encoded representation, as shown in Figure 14.

```
array([[3817, 3975, 4889, ..., 4954, 1462, 1729],
       [1203, 4728,  328, ..., 4954, 1462, 1729],
       [4833, 2651, 1809, ...,  965, 1276, 4231],
       ...,
       [4954, 2898, 3995, ..., 4877, 4603, 4068],
       [1169, 2121,  841, ..., 2843,   57, 4231],
       [4448, 4006, 3846, ..., 1784, 3469, 1378]], dtype=int32)
```

```
] all_labels

array([[1., 0., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 0., 1.]])
```

Figure 14: One Hot Representation for Dataset and Labels

Preparation and Training of the Model: The first step in training any machine learning or deep learning model is to segment the input data into two sets of training and testing and divide the data into an 80%-20% ratio. Initially, we used two models, Logistic Regression with Countvectorizer and Naive Bayes with TF-IDF vectorizer, to handle the classification.

Evaluating the Model: The final stage is to evaluate the classifier model based on its accuracy**.** Accuracy is the percentage of correct predictions over total predictions. Here, we have used a limited news dataset for better results that can be extended to a large data set. The training and testing accuracy scores for binary and multi-class classification are as follows:

Logistic Regression with Count Vectorizer	
Training Accuracy	99.8%
Testing Accuracy	83.3%
Naive Bayes with TF-IDF vectorizer	
Training Accuracy	91%
Testing Accuracy	66%

Table 3: Accuracy of Binary Classification

Logistic Regression with Count Vectorizer	
Training Accuracy	99.9%
Testing Accuracy	57.5%
Naive Bayes with TF-IDF vectorizer	
Training Accuracy	33%
Testing Accuracy	29%

Table 4: Accuracy of Multi-Class Classification

5 CONCLUSION

The research paper is the classification model for Hindi Newswire articles based on NLP and Machine Learning. These classification models only exist for English text. The classification of newswire articles has been divided into two parts: preprocessing the dataset and preparing the classification model. This paper mainly presented the initial requirements for the Dataset preparation and related preprocessing steps. It also provides a comprehensive summary of the results. For preprocessing, news articles have been collected from multiple resources. The work includes cleaning, preprocessing, and converting the cleaned files to the unified format and tagging the news reports as per the categories and sub-categories. Prepared modules for Parts of Speech Tagging, Tokenization, and Keyword Extraction in Hindi. Currently, no available system classifies the Hindi text into predefined categories by considering newswire articles, processing them, and organizing random news articles. We plan to build a classification model using our own Hindi dataset using NLP, Machine, and Deep Learning. This work can also further be expanded into the areas with the Expansion of Hindi Datasets and then the development of a Classification system with an innovative algorithm, providing better performance with outstanding results.

REFERENCES

[1] Sahoo K S, Saha S, Ekbal A, and Bhattacharyya P, 2020 "A platform for Event Extraction in Hindi*" Proceedings of the 12th Conference on Language Resources and Evaluation (LREC)*, pp 2241–2250.

[2] Puri S and Singh P S, 2019 "An Efficient Hindi Text Classification Model Using SVM," *Springer Nature Singapore Pte Ltd*.

[3] Sahoo K S, Saha S, Ekbal A, Bhattacharyya P and Mathew J 2019 "Event-Argument Linking in Hindi for Information Extraction in Disaster Domain" *CICLing 2019*.

[4] Ahmad Z, Sahoo K S, Ekbal A, and Bhattacharyya P 2018 "A Deep Learning Model for Event Extraction and Classification in Hindi for Disaster Domain" *Proc. of ICON*-2018, Patiala, India. December 2018 c2018 NLPAI, pp. 127–136.

[5] Dittrich A and Lucas C 2014. Is this Twitter event a disaster? *AGILE Digital Editions.*

[6] Soni S., Chouhan S.S. and Rathore S.S. 2023 TextConvoNet: a convolutional neural network based architecture for text classification. *Appl Intell* 53, 14249–14268 . https://doi.org/10.1007/s10489-022-04221-9

[7] Pal K and Patel V. B 2020 Modal for classification of Poems in Hindi Language Based on Ras, *Springer Nature Singapore*.

[8] Jain L and Agrawal P, 2019 "Hindi Story Heading Generation Using Proverb Identification" *Springer Nature Singapore Pte Ltd.*

[9] Soni V. K. and Selot S., 2021"A Comprehensive Study for the Hindi Language to Implement Supervised Text Classification Techniques," *6th International Conference on Signal Processing, Computing and Control (ISPCC)*, Solan, India, pp. 539-544, doi: 10.1109/ISPCC53510.2021.9609401.

[10]Sarkar K. 2020 Heterogeneous classifier ensemble for sentiment analysis of Bengali and Hindi tweets. *Sādhanā* 45, 196. https://doi.org/10.1007/s12046-020-01424-z

[11] Kowsari K, Meimandi J K, Heidarysafa M, and Mendu S, Barnes L 2019 "Text Classification Algorithms: A Survey," *Information*, 10, 150; doi:10.3390/info10040150.

[12] Palanivinayagam A, El-Bayeh CZ, and Damaševičius R., 2023, Twenty Years of Machine-Learning-Based Text Classification: A Systematic Review. *Algorithms*., 16(5):236. https://doi.org/10.3390/a16050236

[13] Sharma A and Ghose U 2023 "Toward Machine Learning Based Binary Sentiment Classification of Movie Reviews for Resource Restraint Language (RRL)—Hindi" *IEEE Access* Digital Object Identifier 10.1109/ACCESS.2023.3283461

[14] Devi S J, Bai R M., and Reddy C. 2020."Newspaper Article Classification using Machine Learning Techniques".*International Journal of Innovative Technology and Exploring Engineering*, 9(5), pp. (872-877)

[15] Kumar S and Singh D T. 2022. "Fake news detection on Hindi news dataset.*Global Transitions Proceedings*".2(1), pp. (289-297)

[16] Ali A. M and Kulkarni B. S. 2020 "Preprocessing of Text for Emotion Detection and Sentiment Analysis of Hindi Movie Reviews" *International Conference on IoT based Control Networks and Intelligent Systems* (*ICICNIS* 2020).

[17] Vijayarani S., Ilamathiv J., and Nithya 2020 "Preprocessing Techniques for Text Mining - An Overview" Dr.S.Vijayarani et al , *International Journal of Computer Science & Communication Networks*, 5(1), pp. 7-16, ISSN:2249-5789.

[18] Kadhim I A 2018 "An Evaluation of Preprocessing Techniques for Text Classification" *International Journal of Computer Science and Information Security (IJCSIS)*, 16(6).

[19] Alam S and Yao N 2018 "The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis" Comput Math Organ Theory(2019) 25:319–335 https://doi.org/10.1007/s10588-018-9266-8 *Springer Science+Business Media, LLC, part of Springer Nature* .

[20] Magotra S., Kaushik B. and Kaul A. 2020 A Comparative analysis for identification and classification of text segmentation challenges in Takri Script. *Sādhanā* 45, 146 . https://doi.org/10.1007/s12046-020-01384-4

[21] Kannan S. and Gurusamy V, 2015 "Preprocessing Techniques for Text Mining" https://www.researchgate.net/publication/273127322

[22] Jha V., Manjunath N., Shenoy P. D., and Venugopal K. R., 2016 "HSRA: Hindi stopword removal algorithm," 2016 International Confere nce on Microelectronics, Computing and Communications (MicroCom), Durgapur, India, pp. 1-5, doi: 10.1109/MicroCom.2016.7522593.

[23] Sahu S. S. and Pal S. 2022 "Effect of stopwords Indian language IR" *Sådhanå* 47:17 Indian Academy of Sciences https://doi.org/10.1007/s12046-021-01731-z

[24] Dutta D., Halder S., and Gayen T. 2023. Intelligent Part of Speech tagger for Hindi. *Procedia Computer Science*, 218, pp. 604-611.

[25] Thomas A., Kowar M. K., Sharma S. and Sharma H. R., "Exploring Text Semantics to Extract Key-Fragments for Model Answers," 2010 *International Conference on Advances in Recent Technologies in Communication and Computing*, Kottayam, India, pp. 255-257, doi: 10.1109/ARTCom.2010.110.

[26] Shrivastava M., and Bhattacharyya P. 2008. Hindi POS Tagger Using Naive Stemming : Harnessing Morphological Information Without Extensive Linguistic Knowledge.

[27]Joshi N, Darbari H, and Mathur T. 2013. HMM Based POS Tagger for Hindi. *Third International Conference on Computer Science & Information Technology*: pp. (341-349)

[28] Gadde K S P. and Yeleti V. M. 2008. Improving statistical POS tagging using Linguistic feature for Hindi and Telugu. ICON-2008: *International Conference on Natural Language Processing*: pp. (1-8).

[29] Siddiqi S. and Sharan A., 2015 "Keyword and keyphrase extraction from single Hindi document using statistical approach," 2015 *2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, pp. 713-718, doi: 10.1109/SPIN.2015.7095377.

[30] Mahato S., and Thomas A. 2017 “Lexico-semantic analysis of essays in Hindi language” CEUR Workshop Proceedings , 1819.