

A
PROJECT REPORT
On
HNVec-Driven MultiDL Framework for
Content-Based Classification of Hindi
News Articles

*Submitted in partial fulfillment of the
requirements for the award of the
degrees*

of
BACHELOR OF TECHNOLOGY
in

INFORMATION TECHNOLOGY

Submitted by:

Niladri Ghosh (300103321015)

Aarushi Shrivastava (300103321019)

Devdeep Sarkar (300103321033)

Subhodeep Sarkar (300103321050)

Guided by:

Mrs. K. Subhashini Spurjeon

(Assistant Professor)



BHILAI INSTITUTE OF TECHNOLOGY DURG
DEPARTMENT OF INFORMATION TECHNOLOGY

UGC Autonomous Institution

(Affiliated to CSVTU, Approved by AICTE, NBA & NAAC ACCREDITED)

DURG– 491001, CHHATTISGARH, INDIA
www.bitdurg.ac.in

SESSION: 2024-25

A
PROJECT REPORT
On
HNVec-Driven MultiDL Framework for
Content-Based Classification of Hindi
News Articles

*Submitted in partial fulfillment of the
requirements for the award of the
degrees*

of
BACHELOR OF TECHNOLOGY
in
INFORMATION TECHNOLOGY

Submitted by:

Niladri Ghosh (300103321015)
Aarushi Shrivastava (300103321019)
Devdeep Sarkar (300103321033)
Subhodeep Sarkar (300103321050)

Guided by:

Mrs. K. Subhashini Spurjeon
(Assistant Professor)



BHILAI INSTITUTE OF TECHNOLOGY DURG
DEPARTMENT OF INFORMATION TECHNOLOGY

UGC Autonomous Institution

(Affiliated to CSVTU, Approved by AICTE, NBA & NAAC ACCREDITED)

DURG– 491001, CHHATTISGARH, INDIA
www.bitdurg.ac.in

SESSION: 2024-25

CANDIDATE’S DECLARATION

We hereby declare that the project entitled “**HNVec-Driven MultiDL Framework for Content-Based Classification of Hindi News Articles**” submitted in partial fulfillment for the award of the degree of Bachelor of Technology in Information Technology completed under the supervision of **Mrs. K. Subhashini Spurjeon, Assistant Professor, Information Technology, BIT DURG** is an authentic work.

Further, I/we declare that I/we have not submitted this work for the award of any other degree elsewhere.

Niladri Ghosh

Aarushi Shrivastava

Devdeep Sarkar

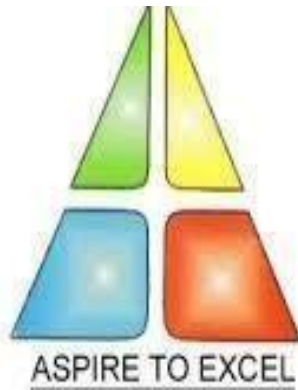
Subhodeep Sarkar

Signature and name of the student(s) with date

CERTIFICATE by PROJECT Guide(s)

It is certified that the above statement made by the students is correct to the best of my/our knowledge.

Signature of BTP Guide(s) with dates and their designation



BHILAI INSTITUTE OF TECHNOLOGY DURG
DEPARTMENT OF INFORMATION TECHNOLOGY

UGC Autonomous Institution
(Affiliated to CSVTU, Approved by AICTE, NBA & NAAC ACCREDITED)
DURG– 491001, CHHATTISGARH, INDIA

Department of Information Technology

CERTIFICATE BY THE EXAMINERS

This is to certify that the Major Project work entitled “**HNVec-Driven MultiDL Framework for Content-Based Classification of Hindi News Articles**” is carried out by **Niladri Ghosh (300103321015)**, **Aarushi Shrivastava (300103321019)**, **Devdeep Sarkar (300103321033)**, **Subhodeep Sarkar (300103321050)** in partial fulfillment for the award of degree of **Bachelor of Technology in Information Technology, Chhattisgarh Swami Vivekanand Technical University, Durg** during the academic year 2024-2025.

Prof. Dr. Ani Thomas

HOD

Internal Examiner

External Examiner

ACKNOWLEDGEMENTS

We wish to acknowledge with a deep sense of hearty gratitude and indebtedness to **Mrs. Babita Verma**, Information Technology, who gave us this opportunity to experience project work & his valuable suggestions during this project have been invaluable.

We take this opportunity to voice & record our sincerest gratefulness towards our esteemed Supervisor **Mrs. K. Subhashini Spurjeon** under whose able guidance the project work has been brought to completion.

Our heart leaps up in thankfulness for his benevolence & time to time help, valuable suggestions, constructive criticism & active interest in the successful completion of this project work.

We are also thankful to all our honorable teachers of the Information Technology Department and our parents whose valuable support helped us and kept us motivated all through.

Niladri Ghosh

Aarushi Shrivastava

Devdeep Sarkar

Subhodeep Sarkar

B.Tech. VIIIth Sem

Discipline of Information Technology

Bhilai Institute of Technology, Durg

Abstract

Automatic news classification has come to be an important subject in Natural Language Processing, especially for low-resource languages like Hindi, due to the exponential growth of digital news material in regional languages. Personalized suggestions, real-time news filtering, and content organization all depend on the effective classification of Hindi news articles into predetermined categories like politics, sports, technology, and entertainment.

This project offers an approach for classifying Hindi news by assessing how well three distinct text embedding strategies—TF-IDF, FastText, and HNVec—work when combined with a **HLM-CLS** hybrid model. TF-IDF and FastText serve as baseline methods, representing frequency-based and subword-aware embeddings, respectively. **HNVec** is introduced as a novel distance based context-sensitive vectorizer tailored to capture the semantic and contextual nuances of Hindi text. The HLM-CLS model is evaluated in combination with each vectorizer to assess their impact on classification performance. The hybrid approach integrates the excellent classification ability of SVM, feature extraction capabilities of CNN, and interpretability of LR machine learning models. The pipeline using HNVec embeddings perform noticeably better than those using TF-IDF and FastText in terms of accuracy, F1 score, precision and recall parameters. The training and validation loss metrics also indicate that HNVec is very effective for deep text classification tasks in Hindi because it allows for stronger semantic representation and faster convergence. The proposed HLM-CLS with HNVec provides the highest 88.61% accuracy, 88.50% precision, 88.61% recall, and 88.51% F1 Score as compared to other combinations.

This project work emphasizes how crucial it is to create linguistically sensitive embeddings for Indian languages and shows how including them into hybrid deep learning models can result in notable gains in classification accuracy. The suggested method creates new opportunities for accurate and scalable NLP applications in languages that are underrepresented.

List of Tables

Table No.	Table Name	Page No.
4.1	Summary of Categorization	28
4.2	NewsWire Dataset Representation	29
4.3	HNVec Dataset Representation	33
5.1	Performance Analysis Table for HLM-CLS with HNVec	49
5.2	Performance Analysis Table for HLM-CLS with TF-IDF	52
5.3	Performance Analysis Table for HLM-CLS with FastText	54

List of Figures

Figure No.	Figure Name	Page No.
3.1	Relation among AI, ML & NLP	11
3.2	Classification of ML models	17
3.3	Graph of Implementation of SVM	18
3.4	Graph of Implementation of LR	19
3.5	Fully Connected Artificial Neural Network	20
3.6	CNN Architecture	21
3.7	Classification of Embeddings	22
4.1	Dataset Before Removing Blank Space and unwanted Lines	27
4.2	Dataset After Removing Blank Space and Unwanted Lines	27
4.3	Classification of Tag News Articles in Hindi	28
4.4	Classification of Tag News Articles in Hindi	28
4.5	Raw Dataset Files	30
4.6	Raw Articles	30
4.7	Articles After cleaning and refining	31
4.8	Final Dataset CSV File	32
4.9	Flow Chart	46
4.10	Architecture of HNVec with HLM-CLS	47
5.1	Confusion Matrix of HLM-CLS with HNVec for 30 Epoch	50
5.2	Training vs Validation Loss of HLM-CLS with HNVec for 30 Epoch	50
5.3	Confusion Matrix of HLM-CLS with TF-IDF for 40 Epoch	53
5.4	Training vs Validation Loss of HLM-CLS withTF-IDF for 40 Epoch	53
5.5	Confusion Matrix of HLM-CLS with FastText for 50 Epoch	55
5.6	Training vs Validation Loss of HLM-CLS FastText for 40 Epoch	55

List of Abbreviations

CNN	Convolution Neural Network
SVM	Support Vector Machine
LR	Logistic Regression
HNVec	Hindi News Vectorizer
NLP	Natural Language Processing
HLM	Hybrid Learning Model
CLS	CNN-LR-SVM
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning

Table of content

Abstract	i
List of Tables	ii
List of figures	iii
List of Abbreviations	iv
Chapter 1 Introduction	1-7
1.1 Overview of HNVec-Driven MultiDL Framework for Content-Based Classification of Hindi News Articles	1
1.2 Significance of Hindi Text Classification	1
1.3 Application Areas of Hindi News Classification	2
1.4 Challenges of Hindi News Classification	5
1.5 Motivation	5
1.6 An Overview of the Thesis	6
1.7 Summary	7
Chapter 2 Review of Literature	8-10
2.1 Introduction	8
2.2 Existing classification system for English articles	8
2.3 Gaps in Existing Research	10
2.4 Summary	10
Chapter 3 Theoretical Background	11-25
3.1 Introduction	11
3.2 NLP in the field of Computer Science	12
3.3 Background of Text Classification System	13
3.4 Machine Learning Model	15
3.4.1 Support Vector Machine	17
3.4.2 Logistic Regression	18
3.5 Deep Learning Model	20
3.5.1 Convolutional Neural Networks	21
3.6 Overview of Vectorizer	22
3.6.1 Context-independent embeddings	23
3.6.2 Context Dependent word embedding	24
3.7 Summary	25

Chapter 4	Methods and Materials	26 – 47
4.2	Employed Dataset	26
4.2.1	Hindi NewsWire Dataset	26
4.2.2	HNVec Dataset	29
4.3	Hybrid Classifier Models Integrated withVectorizer	33
4.3.1	HLM-CLS with HNVec	34
4.3.2	HLM-CLS with TF-IDF	34
4.3.3	HLM-CLS with FastText	35
4.4	HNVec: A Normal Vectorizer for Hindi News Categorization	35
4.4.1	Co-occurrence Matrix	36
4.4.2	HNVec Model Training	37
4.4.3	Algorithm for HNVec with HLM-CLS	40
4.4.4	Pseudo Code for HNVec	43
4.4.5	Flow Chart	46
4.4.6	Architecture	47
4.5	Summary	47
Chapter 5	Results and Discussions	48 - 56
5.1	Introduction	48
5.2	Vectorizer Analysis	48
5.2.1	Analysis of HNVec	48
5.2.2	Analysis of TF-IDF	51
5.2.3	Analysis of FastText	54
5.3	Comparative Analysis	56
5.4	Summary	56
Chapter 6	Summary and Conclusion	57 - 58
6.1	Introduction	57
6.2	Outline of proposed work	57
6.3	Future Scope of Work	58
6.4	Summary	58
REFERENCES		59 - 60

CHAPTER 1

INTRODUCTION

The thesis discusses the Machine Learning and Natural Language Processing techniques to classify Hindi news articles efficiently. The purpose of the thesis is to use the vectorizer along with machine learning and deep learning models and to evaluate the performance of multiclass hybrid models and to analyse a well-structured hybrid approach that appears to be best, considering its high accuracy and less losses. This chapter briefly introduces Hindi news categorization along with its significance, challenges and their potential application areas. We also discuss the motivation of this thesis. Finally, we describe the organization of the thesis and the topics of each chapter.

1.1 OVERVIEW OF MULTIDL HINDI NEWS CATEGORIZATION

Language plays a vital role in communication. There are thousands of dialects and languages spoken throughout the world. Especially, Hindi underwent significant evolution over the past millennium, emerging as a prominent global language today. Its presence on online platforms is crucial, serving as a means of expression for news outlets, government bodies, and various sectors. While Unicode facilitates online reading and writing, challenges persist, notably in Information Extraction and text classification. Text classification is particularly vital due to the abundance of uncategorized data. Leveraging natural language processing and machine learning, Hindi news articles can be effectively analysed and categorized, covering diverse topics such as world affairs, sports, politics, and economy. Machine learning and deep learning algorithms were used to classify the news articles. The classifiers were used and evaluated in terms of accuracy, precision, F1 score, and recall, achieving a better accuracy rate. Various algorithms, including Support Vector Machines, Logistic Regression, the vectorizer, and Convolutional Neural Networks, were used to develop a hybrid model.

1.2 SIGNIFICANCE OF HINDI TEXT CLASSIFICATION

Text classification is a common NLP task used to solve business problems in various fields. The goal of text classification is to categorize or predict a class of unseen text documents, often with the help of supervised machine learning. The classification of Hindi news articles using machine learning is an essential task with several significant aspects. Here's a detailed overview:

1. Accessibility and Information Retrieval:

- Improves accessibility to Hindi content for users by categorizing articles into predefined

categories like sports, politics, or entertainment, aiding efficient retrieval.

- Enables better personalization of content based on user preferences.

2. Regional Language Processing:

- Promotes technological advancements in Indian languages, especially Hindi, which is spoken by millions, encouraging inclusivity in AI-driven solutions.

3. Automation and Scalability:

- Automates the manual task of categorizing large volumes of news data, saving time and resources for media organizations.
- Facilitates handling the continuous inflow of data from various sources like websites and social media.

4. Sentiment Analysis and Opinion Mining:

- Useful in analyzing public sentiment on trending topics.
- Helps policymakers and businesses understand public opinion on socio-political issues.

5. Fake News Detection:

- Acts as a foundational system for building fake news detection frameworks in Hindi, critical for combating misinformation in regional languages.

1.3 APPLICATION OF HINDI NEWS CATEGORIZATION

The Hindi news classification system is a versatile tool that not only improves the accessibility and management of news content but also drives innovation in media, governance, business, and social sectors. Here are detailed application areas:

1. Media and Journalism

Content Organization: Automatically organizes and classifies news articles into predefined categories (e.g., sports, politics, entertainment). Simplifies newsroom operations by reducing manual effort in sorting news stories.

Customized News Delivery: Provides personalized news feeds to users based on their reading habits and preferences. Enables news aggregation platforms to offer curated content.

Real-Time Alerts: Helps media outlets send category-specific notifications to users (e.g., breaking news in politics or sports updates).

2. Search Engines and Information Retrieval

Improved Search Results: Enhances the relevance of search results by categorizing news articles, making them easily searchable by topic.

Semantic Search: Facilitates advanced search capabilities by understanding the context and meaning of queries in Hindi.

3. Social Media Monitoring

Trend Analysis: Identifies trending topics in specific categories (e.g., elections in politics, movie releases in entertainment).

Content Moderation: Helps in filtering and categorizing news content shared on social media platforms to ensure relevance and compliance.

Sentiment Analysis: Extracts public sentiment around news topics for businesses, policymakers, and media organizations.

4. Fake News Detection and Misinformation Control

Misinformation Identification: Serves as a foundation for building fake news detection systems in Hindi by categorizing and analyzing news content.

Fact-Checking: Supports automated fact-checking processes by identifying and tagging suspicious articles for human review.

5. E-Governance and Public Awareness

Government Portals: Helps government websites and applications categorize news articles related to schemes, policies, and updates for easier access.

Public Awareness Campaigns: Enables targeted dissemination of information in specific categories like health or education to reach the Hindi-speaking population.

6. Education and Research

Language Learning Tools: Facilitates the creation of language resources and tools for learning Hindi through classified and structured content.

Academic Research: Provides a base for linguistic and social research on media trends, regional narratives, and public discourse in Hindi.

7. Advertising and Marketing

Targeted Campaigns: Enables marketers to categorize and analyze news articles for identifying suitable platforms for ads.

Content-Based Advertising: Helps deliver advertisements related to specific categories, like sports gear ads on sports news.

8. Business Intelligence

Market Analysis: Allows businesses to understand market trends in various domains by analyzing classified news data.

Competitor Analysis: Enables companies to monitor industry news about competitors in specific categories like technology or finance.

9. News Analytics

Reader Insights: Assists media companies in understanding which categories attract the most readers and tailoring their content strategy accordingly.

Performance Metrics: Tracks the popularity of specific categories over time for strategic decision-making.

10. Legal and Regulatory

Content Compliance: Assists in monitoring news for compliance with legal and regulatory requirements by categorizing and tagging sensitive content.

Hate Speech Detection: Identifies and categorizes news articles to flag potentially harmful or inflammatory content.

11. Regional and Local News Management

Hyperlocal Journalism: Supports local news categorization for better management and distribution of region-specific content.

Diversity in News Coverage: Encourages the inclusion of underrepresented topics by identifying gaps in news coverage across categories.

12. Disaster Management and Crisis Response

Real-Time Information: Helps categorize and distribute critical information during natural disasters, pandemics, or crises (e.g., health, rescue, and recovery updates).

Alerts and Warnings: Enables targeted communication of warnings to affected regions through categorized news.

1.4 CHALLENGES OF HINDI NEWS CATEGORIZATION

1. Linguistic Complexity:

- Hindi has rich morphology, with numerous word forms, synonyms, and a flexible word order, complicating text analysis and feature extraction.
- The presence of regional dialects and mixed-code texts (e.g., Hindi-English) makes preprocessing more difficult.

2. Limited Resources:

- Scarcity of high-quality labeled datasets for Hindi news articles.
- Lack of robust pre-trained language models for Hindi, though models like multilingual BERT partially address this.

3. Vocabulary and Ambiguity:

- Hindi has a large vocabulary, including loanwords from Sanskrit, Urdu, and English, adding complexity to tokenization and vectorization.
- Words may have multiple meanings depending on context.

4. Data Imbalance:

- Categories like "politics" and "entertainment" might dominate over others like "technology," leading to biased models.

5. Preprocessing Challenges:

- Handling non-standard writing styles, informal spellings, and typos commonly seen in user-generated news content.
- Dealing with punctuation, stopwords, and stemming in Hindi.

6. Evaluation Complexity:

- Difficulty in defining evaluation metrics that effectively capture semantic nuances in Hindi text.

7. Computational Costs:

- Training deep learning models on resource-constrained systems can be challenging due to high computational and memory requirements.

By integrating this system, organizations can unlock significant operational efficiencies and better serve the Hindi-speaking population.

1.5 MOTIVATION

The motivation for Hindi news classification stems from several societal, technological, and business-driven factors:

- **Bridging the Language Gap**

Enable millions of Hindi-speaking users to access categorized and relevant news in their native language, overcoming the dominance of English-based systems. Foster technological inclusion for Hindi and other Indian languages in AI applications, promoting linguistic and cultural diversity.

- **Enhancing AI Capabilities for Hindi**

Advance the field of natural language processing (NLP) for Hindi, contributing to the creation of robust language models and tools. Motivate the collection and labeling of high-quality Hindi datasets for future research and applications.

- **Supporting Media and Journalism**

Reduce the manual effort required in categorizing and publishing news articles, allowing journalists to focus on core content creation. Personalize news delivery for readers by offering content tailored to their preferences and interests.

- **Improving Information Retrieval**

Simplify the management of large volumes of Hindi news content by automating categorization into predefined topics. Make news more searchable and accessible for users, especially for specific queries like politics, sports, or entertainment.

1.6 AN OVERVIEW OF THESIS

The thesis describes a novel vectorizer tailored for Hindi News text representation and conducts a broad evaluation of its performance. The proposed vectorizer is compared against four different widely used vectorizers. The thesis is organized into the following chapters:

Chapter 2 gives the complete literature survey of Text categorization research work done in India and foreign countries.

Chapter 3 provides the theoretical background of the research work in which different Machine learning and deep learning models and its application areas are discussed.

Chapter 4 is related to pre-processing phase and different vectorizers integrated with hybrid models used for classification of hindi news. It describes the pre-processing stage, various linguistic tools, and other methodologies.

Chapter 5 gives results and a discussion of the research work. Finally, in Chapter 6, we draw some conclusions and present suggestions for further work.

1.7 SUMMARY

This chapter introduces. It is followed by application areas of machine translation, challenges of the machine translation system, and motivation. The chapter concludes with a brief outline of the thesis. The next chapter provides a survey of the existing literature in the field of machine categorization.

CHAPTER 2

REVIEW OF LITERATURE

2.1 INTRODUCTION

Hindi news categorization is an advanced application in natural language processing that focuses on efficiently classifying Hindi news articles into predefined categories. This approach leverages a hybrid model where various techniques and models are combined to enhance efficiency and achieve highly accurate results. In this chapter, a complete survey has been conducted to identify the problem statement. The survey work is divided into two parts. In the first part, research work already conducted in the domain of text classification is reviewed, with a particular focus on existing classification systems for English articles. In the second part, gaps in the existing research are identified. This chapter is structured into two sections: Section 2.2 discusses existing classification systems for English articles, while Section 2.3 highlights the gaps in current research with respect to Hindi news categorization.

2.2 EXISTING CLASSIFICATION SYSTEM FOR ENGLISH ARTICLES

Vipin et al. proposed a hybrid CNN-LSTM and LSTM-CNN models for sentiment classification into positive, neutral, and negative categories, achieving promising results. [A]

Parthiban et al. compared classifiers like LSTM, DNN-BiLSTM, LR, CNN, RF, and RNN across seven datasets, with the DNN-BiLSTM model outperforming others, achieving good accuracy on the combined Enron dataset. [B]

Ratnam et al. proposed a hybrid model incorporating RNNs and autoencoders was designed, with RNNs capturing document sequential dependencies and autoencoders enhancing feature representation. To improve document clustering, a hybrid model combining RNNs and autoencoders was developed, leveraging the strengths of both architectures for more efficient clustering. [C]

Livieris et al. introduced a two-level ensemble meta-learning strategy that fuses baseline classifier committees. The key idea is to improve performance by enhancing classifier diversity, ensuring the ensemble captures diverse data patterns for more accurate results. According to Livieris et al., the averaging method is the most straightforward approach for combining the predictions of several models. [D][E].

Alzoubi et al. delved into text classification using five different methods and meticulously examined the results. They explored the impact of data preparation by assessing both raw and pre-processed

data. Techniques such as morphological examination, standardization, and simplification were employed on the data. The study compared the effectiveness of approaches including Naive Bayes (NB), Support Vector Machines (SVM), Long Short-Term Memory (LSTM), Logistic Regression (LR), and Random Forest (RF). LSTM emerged as the most efficient method in terms of training time and accuracy, highlighting the significance of coherence between training set size, categories, and normalized data. Additionally, the choice of feature extraction method, particularly TF-IDF, significantly influenced the outcomes, with data simplification demonstrating the most significant impact among all preparation steps. [F]

McCallum et al. discussed advanced methods for selecting important features in text classification, such as mutual information and information gain. They highlighted Naive Bayes as a popular algorithm for such tasks but noted its data sparsity issue. Yin et al. (2017) emphasized the growing prevalence of deep neural networks (DNNs), particularly Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), in natural language processing (NLP) tasks due to their strong expressive capability and reduced need for feature engineering. [G]

Umer et al. emphasized the effectiveness of Convolutional Neural Network (CNN) models in the classification of both short and long texts. They highlighted the utility of CNNs in scenarios where class distribution is balanced, owing to their ability to deliver comparable performance while maintaining computational efficiency. This finding underscores the practicality of CNNs as a reliable solution for various text classification tasks. [H]

M. Arora et al. conducted an in-depth exploration of three distinct feature engineering techniques—CountVectorizer, TF-IDF, and Word2Vec—for preprocessing data used in training machine learning algorithms. Through extensive experimentation, they determined that the combination of Multinomial Naïve Bayes with CountVectorizer offered the highest accuracy among the methods evaluated. This finding underscores the importance of choosing appropriate feature engineering techniques for optimizing machine learning model performance. [I]

Sitaula, C. et al. introduced a hybrid feature approach for tweet classification, combining syntactic information from bag-of-words with semantic data from fastText and domain-specific methods. They proposed a multi-channel CNN (MCNN) model to capture multi-scale information, improving sentiment classification into positive, neutral, and negative classes. Domain-specific results vary depending on the parameters and features used. Since text classification relies heavily on semantic and sentiment information and dataset size, achieving consistently high accuracy across different domains

using the same techniques is challenging. [J][K][L]

2.3 GAPS IN EXISTING RESEARCH

Previous research on text classification often overlooked multi-class classification, primarily focusing on single or dual combinations of machine learning and deep learning algorithms. Furthermore, most existing models are designed to classify articles into a limited number of categories, typically ranging from 2 to 3 classes. Additionally, most analyses have been conducted in English, with few systems available for Indian languages. Addressing these gaps is essential for advancing research in text classification, enabling the development of models that are versatile, inclusive, and effective for the Hindi language.

2.4 SUMMARY

Finally, after conducting a comprehensive literature survey, the following objectives have been identified for this research work:

- a) To develop a robust multi-class classification model that takes Hindi news articles as input and predicts one of five possible categories.
- b) To automate the process of classifying Hindi news articles, reducing manual effort and increasing efficiency.
- c) To design and implement a hybrid model for precise categorization of Hindi news articles into the identified categories.
- d) To address challenges such as the inefficiency of manual segregation, the scarcity of effective models for Hindi news classification, and to establish a foundation for future advancements in this domain.

After identifying these research objectives through the literature review, their implementation is carried out by exploring advanced concepts of machine learning, hybrid model techniques, and a detailed analysis of Hindi news article structures, which is discussed in Chapter 3.

CHAPTER 3

THEORETICAL BACKGROUND

3.1 INTRODUCTION

In the previous chapter, research objectives are already derived through a literature review. The theoretical background chapter is an essential chapter of any research work. It incorporates all vital factors, parameters, and theoretical concepts required for research work that are used to implement research objectives. In computer science, machine learning (ML), natural language processing (NLP), and artificial intelligence (AI) are related but distinct technologies. Machine Learning and Natural Language Processing are important subfields of Artificial Intelligence that have gained prominence in recent times. Combined with machine learning algorithms, NLP generate system that learn to perform task on their own and get better through experience.

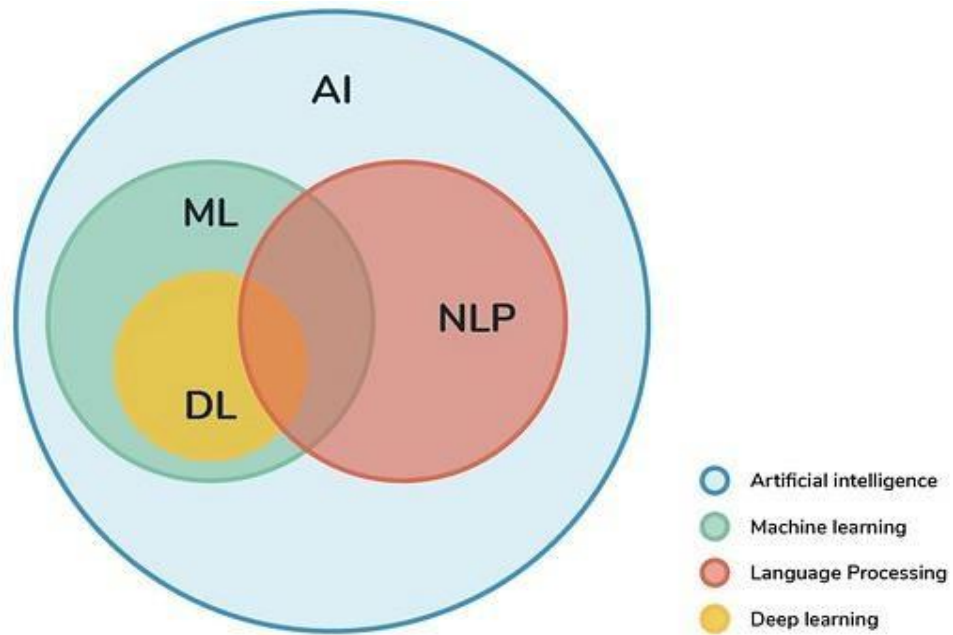


Figure 3.1: Relation among AI, ML & NLP

This chapter is organized into Section 3.2, which is about the use of NLP in the field of Computer Science; section 3.3 discusses machine translation and its different approaches; at the last Section 3.4 discusses the Linguistic background of Hindi and Chhattisgarhi languages.

3.2 NLP IN THE FIELD OF COMPUTER SCIENCE

Natural Language Processing has a subfield called computational linguistics concerned with natural language processing. The computer system converts the information from a database into human-readable form NLP deals with both speech and text. Still, speech processing is a different area to work in as it requires voice samples instead of text.

In NLP, different techniques are required to understand the commands given in human language so that the system can work according to them. To communicate with computers, we need formal languages like Java, C, C++, Python, etc. Understanding these languages is a tedious task and requires a lot of effort. So, this is their limitation in communicating with computers. Compared to this, communication in natural language with the computer system is more straightforward. (Bharti *et al.*, 1994) discussed that Natural Language Processing is an essential subfield of artificial intelligence as the computer will be considered intelligent if it can understand commands in natural language. Natural language processing can be necessary for various applications like machine translation, text summarization, and question-answering systems. The initial processing of natural language is more straightforward, but as we go deep into processing, the difficulty increases in understanding natural language. With the increased ability of computer processing, intelligent machines are more in demand.

NLP uses machine learning to enable a machine to understand how humans communicate with one another. It also leverages datasets to create tools that understand the syntax, semantics, and the context of a particular conversation. Today, NLP powers much of the technology that we use at home and in business.

Natural Language Processing (NLP) and Machine Learning (ML) are synergistically used in text classification to transform raw text into structured and actionable insights. NLP focuses on processing and understanding human language by breaking down text into tokens, removing stop words, performing stemming or lemmatization, and converting text into numerical representations such as Term Frequency-Inverse Document Frequency (TF-IDF), GloVe, Word2Vec, or advanced embeddings. These representations capture semantic and syntactic features of the text, making it suitable for ML models. Machine learning, on the other hand, applies statistical algorithms such as Logistic Regression, Support Vector Machines (SVM), or deep learning architectures like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to identify patterns and classify the text into predefined categories. Together, NLP handles the preprocessing and feature extraction, while ML models use these features to learn and make predictions, enabling applications

like sentiment analysis, spam detection, topic classification, and more. This combination bridges the gap between unstructured language data and computational decision-making, creating robust and scalable text classification systems.

3.3 BACKGROUND OF TEXT CLASSIFICATION SYSTEM

Text classification systems have evolved over decades, driven by the need to manage and analyze large volumes of textual data efficiently. From simple rule-based systems to modern deep learning techniques, the field has undergone significant transformations. Here's a detailed overview of the background:

1. Rule-Based Systems

Early text classification systems were rule-based, relying on manually crafted rules for categorizing text. These rules were typically based on keywords, patterns, or specific linguistic constructs (e.g., presence of words like "sports" or "politics").

Limitations: Highly dependent on domain expertise and manual effort. Poor scalability and adaptability to dynamic data.

2. Statistical Methods

Introduction of Probabilistic Models: Statistical approaches like Naïve Bayes emerged in the late 20th century, leveraging probabilities to predict the category of a text. These methods relied on term frequencies and conditional probabilities to classify text.

Vector Space Models: Representing text as vectors using methods like Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) allowed machines to quantify textual information.

Advantages: Automated the classification process. Scaled better than rule-based systems.

Limitations: Could not capture the semantic meaning of text. Struggled with polysemy (words with multiple meanings) and synonymy (different words with similar meanings).

3. Machine Learning Era

Supervised Learning Models: Models like Support Vector Machines (SVM), Logistic Regression, and Decision Trees became popular for text classification. These methods required labeled training data and performed well when combined with feature engineering techniques like TF-IDF or n-grams.

Feature Engineering: Techniques like n-grams, stemming, lemmatization, and stopwords removal were used to preprocess text and create input features.

Advantages: Improved accuracy and robustness compared to rule-based and statistical methods.

Challenges: Heavily reliant on feature engineering. Struggled to capture deeper semantic relationships and context.

4. Rise of Embedding Techniques

Introduction of Word Embeddings: Word2Vec, GloVe, and FastText revolutionized text classification by representing words as dense, continuous vectors in a semantic space.

These embeddings captured the contextual similarity between words, enabling better classification.

Advantages:

Reduced the need for manual feature engineering. Improved handling of polysemy and synonymy.

Challenges:

Could not capture contextual variations effectively (e.g., the meaning of a word depending on the sentence).

5. Deep Learning Era

Neural Network Models: Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) brought end-to-end learning to text classification. CNNs were effective in identifying n-gram patterns, while RNNs (e.g., LSTMs, GRUs) captured sequential dependencies in text.

Pre-Trained Models: Transfer learning became a game-changer with pre-trained models like ELMo, ULMFit, and eventually, transformer-based models like BERT and GPT.

Advantages: Eliminated the need for extensive feature engineering. Achieved state-of-the-art results across various text classification tasks.

Challenges: Computationally expensive. Required large labeled datasets for fine-tuning.

6. Transformer Models and Beyond

Transformer Architecture: The introduction of the Transformer architecture, and models like BERT, RoBERTa, and GPT, brought unprecedented advances in text classification.

These models are pre-trained on massive corpora and fine-tuned on specific tasks, making them highly versatile.

Contextual Understanding: Bidirectional processing (e.g., in BERT) allowed models to understand the context of a word in a sentence more accurately.

Applications: Used for sentiment analysis, topic classification, spam detection, and more.

Challenges: High resource requirements for training and fine-tuning. Need for large labeled

datasets for specialized tasks.

7. Current Trends

Multilingual Models: Models like mBERT and IndicBERT support multiple languages, enabling text classification in regional and low-resource languages.

Hybrid Models: Combining traditional machine learning methods (e.g., SVM, Logistic Regression) with embeddings from pre-trained models.

Few-Shot and Zero-Shot Learning: Emerging approaches like GPT-4 allow classification with minimal labeled data, making them suitable for low-resource scenarios.

Focus on Low-Resource Languages: Increasing emphasis on developing datasets, embeddings, and models for languages like Hindi to democratize AI adoption.

The journey of text classification systems reflects the broader evolution of artificial intelligence, from manual and rule-based methods to deep learning and transformer-based solutions. Each phase has contributed tools and techniques that address specific challenges, and today's systems are capable of understanding and categorizing text with remarkable accuracy. However, challenges like resource intensity, handling mixed languages, and low-resource scenarios continue to drive research in this field.

3.4 MACHINE LEARNING MODEL

Machine Learning is the branch of research that enables computers to learn without being explicitly programmed. ML is one of the most interesting technologies to have ever come across. Machine learning use algorithms to evaluate enormous volumes of data, detect patterns, and anticipate outcomes. The algorithms improve with time as they are taught on more data. The learning system of a machine learning algorithm is divided into three major parts:

Decision Process: Machine learning techniques are typically used to predict or classify data. Based on some labeled or unlabeled input data, your algorithm will generate an estimate of a pattern in the data. An error function examines the model's predictions. If there are known examples, an error function can compare them to determine the model's accuracy. A Model Optimization Process: If the model fits better to the data points in the training set, weights are modified to close the gap between the known example and the model prediction. The algorithm will repeat this repeated "evaluate and optimize" process, automatically updating weights until a certain level of accuracy is

reached. Machine learning can be broadly classified into three types based on the nature of the learning system and the data available:

Supervised learning:

In this method, the model is trained using a labeled dataset. In other words, the data is accompanied by a label, which the model attempts to predict. This could range from a category name to a real-valued number. During the training process, the model learns a mapping from the input (features) to the output (label). Once trained, the model can anticipate the output of fresh, previously unknown data. Supervised learning methods are commonly used for regression and logistic regression issues, as well as support vector machines for classification.

Unsupervised learning:

This method of learning is frequently used for clustering and dimensionality reduction. Clustering is grouping related data points together, whereas dimensionality reduction entails lowering the number of random variables under consideration by identifying a collection of primary variables. Common unsupervised learning algorithms include k-means for grouping and Principal Component Analysis (PCA) for dimensionality reduction.

Reinforcement Learning:

In reinforcement learning, the algorithm learns actions for a given set of states that lead to a desired state. It is a feedback-based learning paradigm that collects feedback signals after each state or action by interacting with its surroundings. This input serves as a reward (positive for excellent actions and negative for bad actions), and the agent's purpose is to maximize the positive rewards in order to enhance their performance. The model's behavior in reinforcement learning is similar to that of humans, who learn by experiencing feedback and interacting with their surroundings.

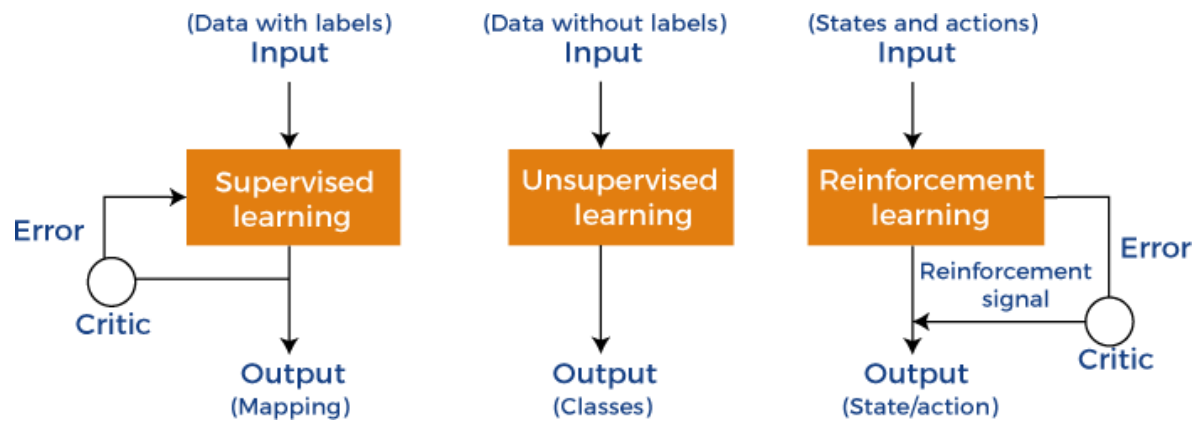


Figure 3.2: Classification of ML models

3.4.1 Support Vector Machine

Support Vector Machine (SVM) is a strong supervised machine learning technique that is commonly used for classification problems. It works by determining the best hyperplane for separating data points from distinct categories with the widest margin. SVMs are often utilized for classification challenges. They discriminate between two classes by determining the best hyperplane that maximizes the difference between the closest data points from opposite classes. The number of features in the input data determines whether the hyperplane is a line in 2D or a plane in n-dimensional space.

Since there are several hyperplanes that can be used to distinguish between classes, the method can determine the optimal decision border between classes by maximizing the margin between points. As a result, it can effectively generalize to fresh data and generate precise categorization forecasts. Since they pass through the data points that establish the greatest margin, the lines that are next to the ideal hyperplane are referred to as support vectors.

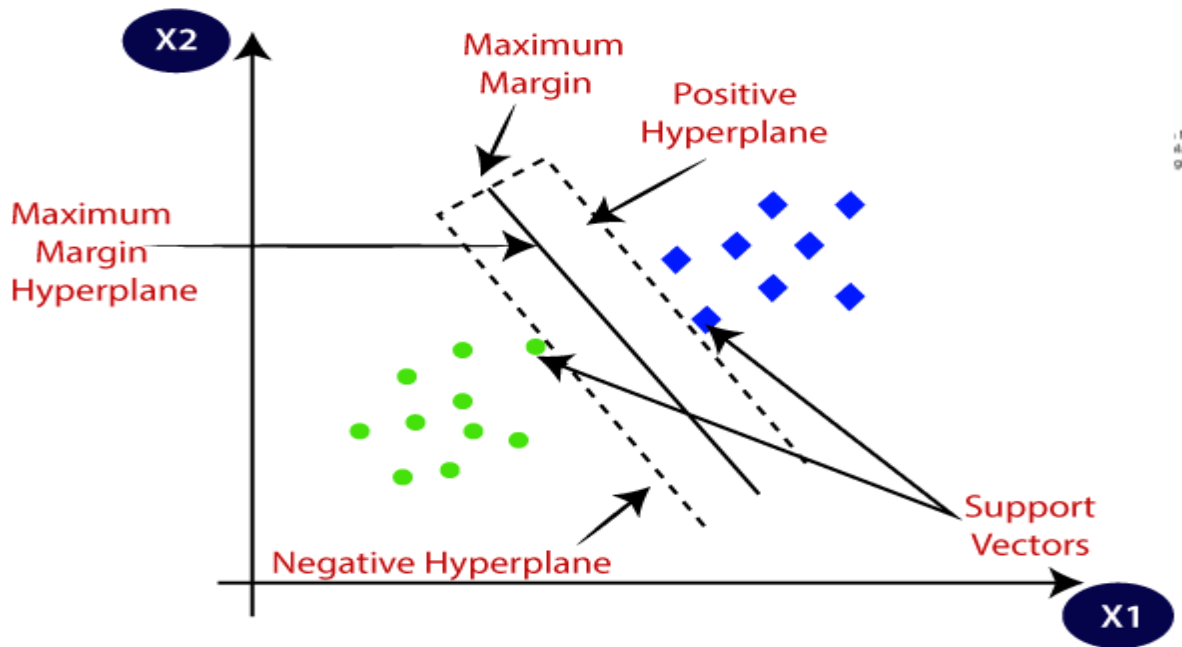


Figure 3.3: Graph of Implementation of SVM

SVM works well for categorizing text into pre-established groups like politics, sports, or entertainment in the case of Hindi news. To use as inputs for the SVM model, Hindi news articles are first transformed into numerical representations using methods like Word2Vec embeddings or Term Frequency-Inverse Document Frequency (TF-IDF). When the data is properly preprocessed, the method produces reliable results and is especially helpful for smaller datasets. It excels at handling high-dimensional data, such as text. Hindi text is preprocessed and transformed into numerical vectors using methods like TF-IDF or Word2Vec since SVM needs numerical inputs. After conversion, SVM groups the articles into pre-established groups (sports, politics, etc.).

3.4.2 Logistic Regression

A statistical model called LR is applied to classification tasks, especially those that involve binary or multiclass problems. The logistic function is used to predict the probability of each category, and the label with the highest probability is assigned. Given a data set of independent variables, logistic regression calculates the likelihood that an event—like voting or not—will occur. The dependent variable has a 0–1 bound because the outcome is a probability.

In logistic regression, a logit transformation is done to the odds, which are the likelihood of

success divided by the probability of failure. Logistic regression maps predictions and probabilities using a logistic function known as the sigmoid function. The sigmoid function is an S-shaped curve that takes any real value and converts it to a range of 0 and 1. Furthermore, if the output of the sigmoid function (estimated probability) exceeds a predetermined threshold on the graph, the model predicts that the instance belongs to that category. If the estimated probability falls below the predefined threshold, the model concludes that the instance does not belong to the class.

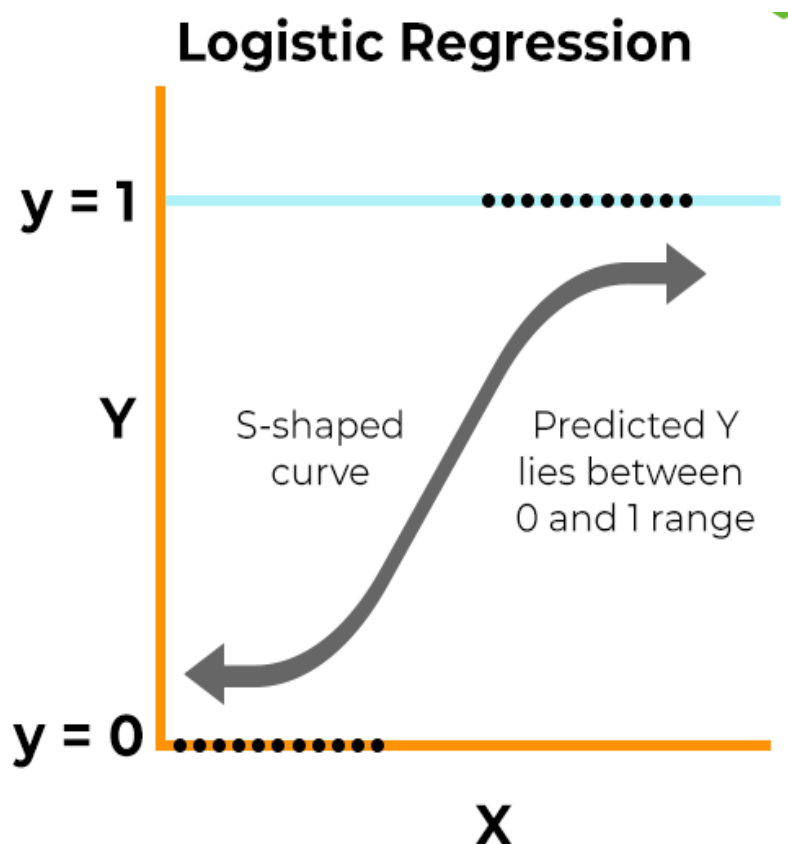


Figure 3.4: Implementation of LR

Logistic Regression can classify Hindi news articles into categories such as entertainment, politics, and technology. The model accepts pre-processed text data, which is frequently translated into numerical formats such as TF-IDF vectors or embeddings. Logistic Regression is simple, interpretable, and efficient for linearly separable data, making it ideal for text classification problems where computational simplicity is important.

3.5 DEEP LEARNING MODEL

Deep learning is defined as a branch of machine learning that uses artificial neural network design. A completely linked Deep neural network consists of an input layer and one or more hidden layers that are connected one after another. Each neuron receives information from either the previous layer's neurons or the input layer. The output of one neuron becomes the input to other neurons in the network's next layer, and so on until the network's output is produced by the final layer. The neural network's layers apply a sequence of nonlinear changes to the input data, allowing the network to learn complicated representations of the data.

Artificial neural networks are based on the structure and function of human neurons. It's also referred to as neural networks or neural nets. Deep learning's most popular architectures are feedforward neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs).

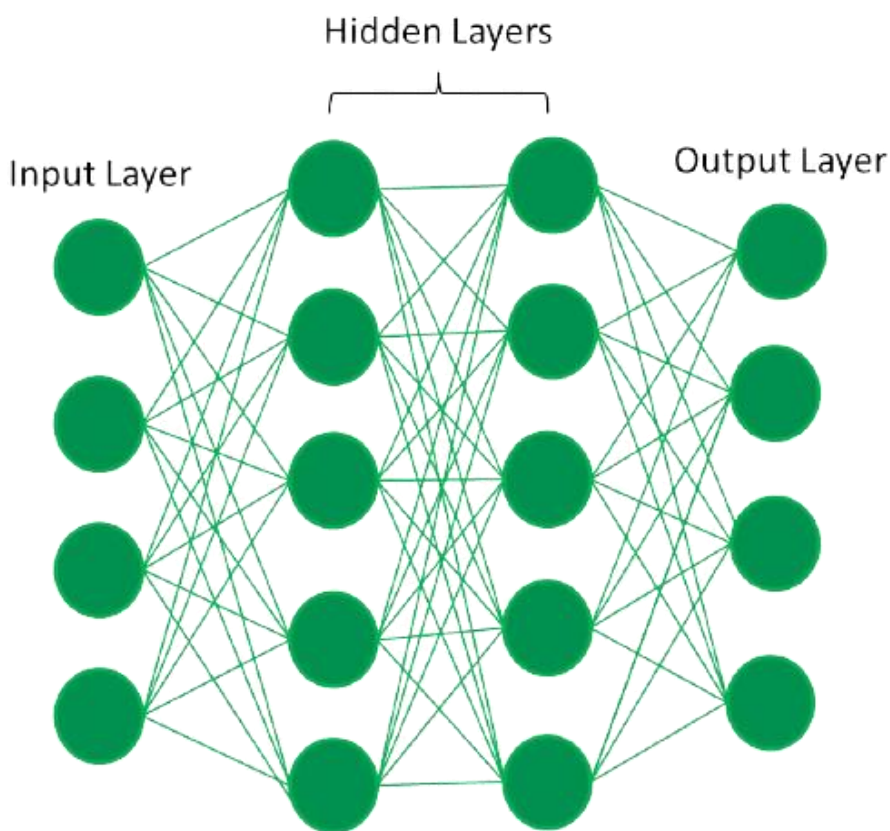


Figure 3.5: Fully Connected Artificial Neural Network

3.5.1 Convolutional Neural Networks

A Convolutional Neural Network (CNN), commonly known as ConvNet, is a form of deep learning algorithm that is specifically built for tasks that require object recognition, such as picture categorization, detection, and segmentation. CNNs, which were originally created for image processing, are also good at text classification. They extract features from text by learning patterns and applying convolutional filters, which allows them to capture local dependencies and n-gram associations. CNNs are used in a range of practical applications, including driverless vehicles, security camera systems, and more. CNN consists of the following parts:

Convolutional layer: This layer employs filters, sometimes known as kernels, to convolve with input images in order to extract significant features and patterns. The convolutional layer learns how to recognize edges, textures, forms, and other visual components.

Pooling layer: This layer decreases the dimension of the feature map via aggregation procedures. This minimizes the memory required to train the network.

Fully connected layer: This layer connects all inputs to a specific set of output neurons. It often correlates to the final layer of a CNN.

Output layer: This is the final layer of the CNN, and it generates the network's predictions. The amount of neurons in this layer is dependent on the task.

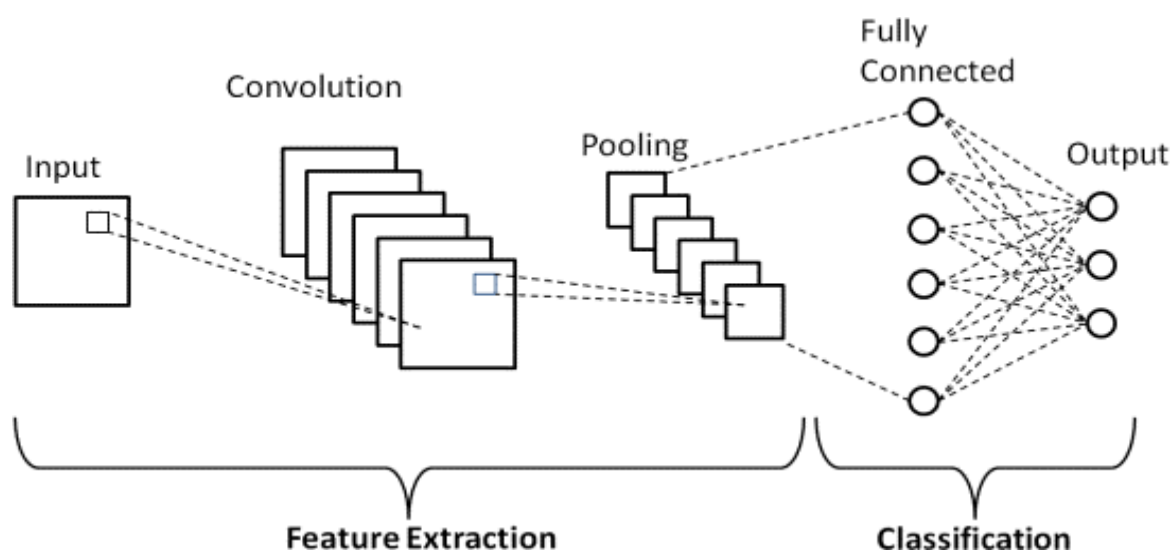


Figure 3.6: CNN Architecture

In Hindi news categorization, text is first transformed into numerical vectors, such as Word2Vec or BERT embeddings, before being fed into the CNN. The model processes these embeddings through convolutional layers to identify meaningful features and patterns that distinguish different categories. The final classification is done using fully connected layers that output probabilities for each category. CNNs are particularly useful for large datasets and can automatically learn hierarchical feature representations from text.

3.6 OVERVIEW OF VECTORIZER

Vectorization is the process of converting text data into numerical vectors. As vectorized representations of textual data, word embeddings are an essential part of pipelines for Natural Language Processing (NLP). They use continuous vector spaces to encode syntactic and semantic information about words. Context-independent and context-dependent embeddings are the two broad categories into which word embeddings fall.

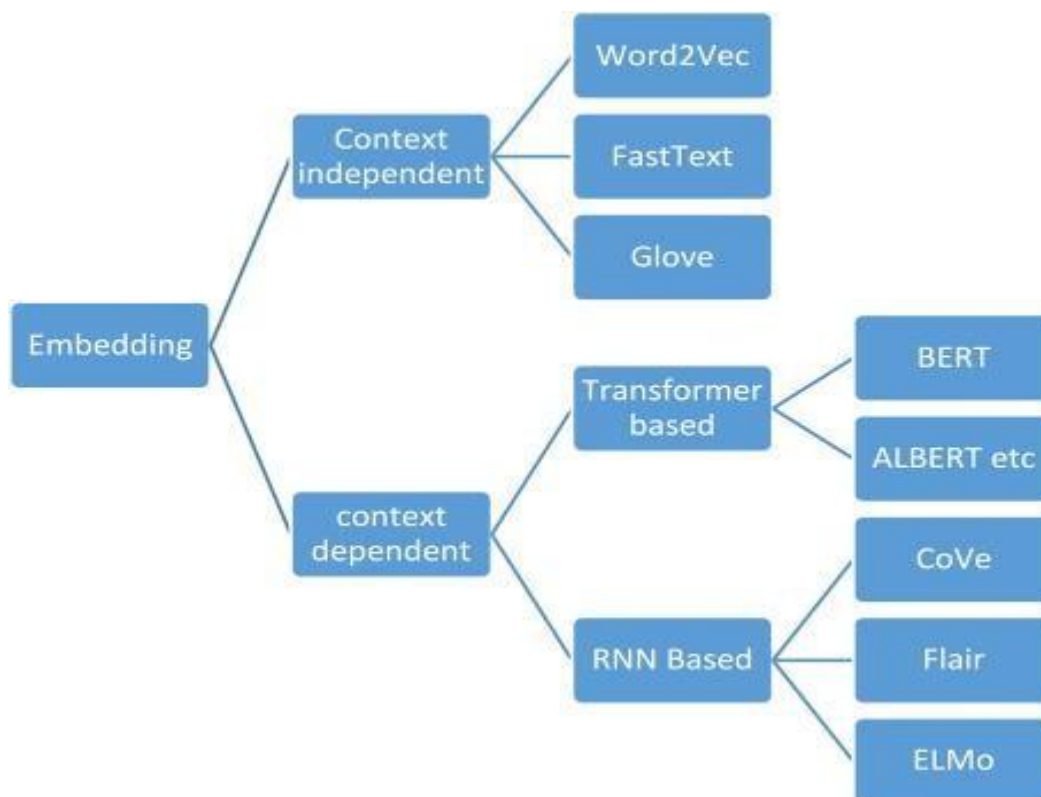


Figure 3.7: Classification Of Embeddings

3.6.1 Context-independent embeddings:

These embeddings create a single fixed representation for every word, independent of the sentence structure or words that surround it. Global statistical semantics are captured by these models, but usage-based meaning variations are not encoded.

- **Word2Vec:** Word2Vec uses two architectures of shallow neural networks: Skip-Gram and Continuous Bag of Words (CBOW). By using local co-occurrence windows to predict a word from its context or vice versa, it learns word representations.

Advantage: It captures both semantic and syntax and also provides contextually rich embeddings.

Disadvantage: Its training and inference are slow due to sequential nature and ignore global co-occurrence.

- **Glove (Global Vectors):** GloVe creates word embeddings by factorizing a co-occurrence matrix and utilizing global co-occurrence statistics. It serves as a link between predictive models and matrix factorization.

Advantages: Both local and global co-occurrence data are encoded and effectively conveys linkages and comparisons.

Disadvantages: Each word has a fixed embedding; it is nevertheless context-independent and lacks contextual subtleties and is plagued by the curse of polysemy.

- **FastText:** In order to produce vectors for out-of-vocabulary (OOV) words, Facebook AI's FastText enhances Word2Vec by modeling words as the sum of character n-gram vectors.

Advantages: strong to OOV and uncommon terms and manages subword information, which is advantageous for languages with complex morphology, such as Hindi.

Disadvantages: Although subword information is modeled, the embeddings remain static and lack context awareness.

- **TF-IDF:** TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents (known as a corpus). It helps convert text data into numerical form, making it suitable for machine learning algorithms.

Advantage: Relevance-Focused as well as easy to compute and integrate into traditional

machine learning pipelines.

Disadvantage: Context Ignorance (TF-IDF does not capture the semantic relationships between words or their contextual meanings) and Dimensionality Issues means for large corpora, the number of unique terms can result in very high-dimensional feature spaces.

3.6.2 Context Dependent word embedding:

These vectorizer solve polysemy and enhance performance in complex natural language processing tasks by dynamically producing distinct representations for a word depending on its environment.

- **Embeddings Based on RNNs (ELMo):** Bi-directional Long Short-Term Memory (BiLSTM) networks are used by ELMo (Embeddings from Language Models) to produce embeddings depending on the context of complete sentences.

Advantages: It offers embeddings that are rich in context. Also captures meaning as well as syntax.

Disadvantages: The sequential nature of RNNs causes slow training and inference and is less parallelizable and resource-intensive than Transformers.

- **Embeddings Based on Transformers:** These include BERT, RoBERTa, etc. it is a Self-attention mechanism used by BERT (Bidirectional Encoder Representations from Transformers) and its variations to model contextual dependencies throughout the sequence.

Advantages: The cutting edge contextual embeddings and excels in tasks like named entity identification, question answering, and sentiment analysis.

Disadvantage: High memory and processing expenses and requires fine-tuning for certain jobs and extensive pre-training corpora.

- **Context Embeddings Based on Distance:** The GloVe paradigm is altered by distance- based context embeddings such as HNVec, which add proximity-weighted co-occurrence data. More weight is given to words that are closer together in a sentence, resulting in context-sensitive embeddings that are still computationally possible.

Advantages: It properly captures spatial context and is more effective than deep neural networks. Also ideal for languages with a rich morphology and free word order.

Disadvantage: lacks strong semantic comprehension and is restricted to shallow models. In contextual tasks with finer details, it might not perform as well as BERT

3.7 SUMMARY

Machine Learning and Natural Language Processing are important subfields of Artificial Intelligence that have gained prominence in recent times. Combined with machine learning algorithms, NLP generate system that learn to perform task on their own and get better through experience. Natural Language Processing (NLP) and Machine Learning (ML) are synergistically used in text classification to transform raw text into structured and actionable insights.

Text classification systems have evolved over decades, driven by the need to manage and analyze large volumes of textual data efficiently. From simple rule-based systems to modern deep learning techniques, the field has undergone significant transformations.

CHAPTER 4

METHODS AND MATERIALS

4.1 INTRODUCTION

The chapter above discusses various machine learning and deep learning models and algorithm. In this section, we discuss the methods implemented to carry out the various activities in the dataset cleaning and preprocessing stage and analyze the machine learning model for classifying Hindi news articles. This section describes the methodology used, which includes two essential elements: building the hybrid learning architecture and thoroughly preparing the dataset. This chapter is organized into Section 4.2, Section 4.3, and Section 4.4. Section 4.2 deals with pre-processing phase and Section 4.3 discusses the Hybrid Classifier Models integrated with vectorizer; in Section 4.4 conclusion is made.

4.2 Employed Dataset

One of the most essential elements of NLP and machine learning is data. The System requires News Articles from Various Categories to Implement the Classification Model. In order to handle the linguistic complexity of Hindi, the dataset is subjected to extensive preprocessing, which includes morphological refinement, stopword removal, tokenization, and normalization. Two datasets are utilized by several vectorizers.

4.2.1 Hindi NewsWire Dataset: These data were gathered from GitHub and Kaggle's BBC Hindi News Dataset. Some of the data were also extracted from the IIT Patna Disaster dataset. This dataset is typically utilized for more than just training. A single training data set that has previously been processed is usually divided into many segments to assess how well the model was trained. Typically, a testing data set is kept apart from the data for this particular reason. The overall dataset is about 3000 articles, and approximately 20 MB of data was used to train and test the models.

Text Pre-processing and feature extraction:

Pre-processing raw data is a crucial step in data preparation, improving the accuracy and efficacy of a machine learning model. To get the dataset ready for the classification model to be applied, the following procedures were taken:

- i) Dataset Cleaning: Cleaning a dataset is the process of eliminating extraneous and noisy data. The main objective is to create a uniform format for the dataset. The data was cleaned by following the steps: Cleaning the dataset started with deleting undesired, irrelevant, and empty ad lines. There are no longer any blank lines in the article; the advertisement lines have been removed, and multi-row

articles have been converted into one row. The next step is to enclose each article within double quotes.

2188	entertainment	"पिछले हफ्ते रिलीज़ हुई उनकी फ़िल्म 'आर.राजकुमार' को बॉक्स ऑफ़िस पर दर्शकों का अच्छा समर्थन मिला. (कहां घिर गए शाहिद कपूर) फ़िल्म व्यापार विशेषज्ञों के मुताबिक़ फ़िल्म ने पहले सप्ताहांत में करीब 31 करोड़ रुपए का कारोबार किया. उम्मीद की जा रही है कि फ़िल्म पहले सप्ताह में ही अपनी लागत वसूल कर लेगी. शाहिद की इस फ़िल्म को भी समीक्षकों ने बकवास करार दिया था लेकिन दर्शकों ने फ़िल्म को पसंद किया. शाहिद के लिए ये राहत की ख़बर है क्योंकि कुछ दिनों पहले रिलीज़ हुई उनकी फ़िल्म 'फटा पोस्टर निकला हीरो' प्रलोप हो गई थी. बॉलीवुड के 'दबंग' और 'मास्टर ब्लास्टर' सचिन तेंदुलकर एक साथ नज़र आने वाले हैं. किसी फ़िल्म में नहीं बल्कि सेलेब्रिटी क्रिकेट लीग यानी सीसीएल के चौथे संस्करण के लॉन्च के मौक़े पर. इसमें सलमान के छोटे भाई सोहेल की टीम भी शामिल है. ये कार्यक्रम 20 दिसंबर को मुंबई के एक फ़ाइव स्टार होटल में आयोजित होगा. सीसीएल-4 की शुरुआत 25 जनवरी से होगी. इस टूर्नामेंट में विभिन्न फ़िल्मी कलाकारों की टीमों शामिल होंगी. हाल ही में सुप्रीम कोर्ट ने भारत में समलैंगिकता को अपराध घोषित किया है. इस सिलसिले में बॉलीवुड ने समलैंगिकों के पक्ष में आवाज़ उठाई है. सुपरस्टार आमिर ख़ान ने कहा, "मैं बहुत ही निराश हूं. ये फ़ैसला मानवाधिकारों का उल्लंघन है. ये बेहद शर्मनाक बात है." अभिनेता-निर्देशक फ़रहान अख़्तर ने कहा कि सुप्रीम कोर्ट का फ़ैसला ग़लत है. वहीं अभिनेत्री श्रुति हासन ने ट्वीट किया, "ये बात सोच के ही कितनी डरावनी लगती है कि कोई और ये फ़ैसला करे कि हमें किससे प्यार करना चाहिए. यानी अपना साथी चुनने की आज़ादी ही ग़ैरकानूनी घोषित कर दी गई है." करण जोहर और ओनीर ने भी सुप्रीम कोर्ट के फ़ैसले पर निराशा जताई. अभिनेत्री अनुष्का शर्मा ने भी इस फ़ैसले को आज़ादी पर हमला बताया. (बीबीसी हिंदी के टर पर भी फ़ॉलो कर सकते हैं.)"
2189	news	सत्रह साल के मोहम्मद समीउल्लाह दक्षिण के शहर कराची में क़ैद हैं. उन पर आरोप है कि उसने एक इम्तिहान के दौरान पेगम्बर मोहम्मद पर अभद्र टिप्पणी की. ह्यूमन राइट्स वॉच ने इस पूरे मामले को 'स्तब्ध करनेवाला' बताया है. पिछले साल नवंबर में एक ईसाई महिला आसिया बीबी को हुई सज़ा के बाद ईश निंदा क़ानून चर्चा में रहा है. हालांकि आसिया बीबी पेगम्बर मोहम्मद के शान में किसी गुस्ताख़ी की बात से इनकार करती हैं. इस साल जनवरी में ही पंजाब के गवर्नर सलमान तासीर की हत्या कर दी गई थी. पुलिस के अनुसार हत्या करनेवाले सलमान तासीर के अंगरक्षक ने कहा कि उसने ऐसा इसीलिए किया क्योंकि तासीर ईश निंदा क़ानून का विरोध कर रहे थे. भय का माहौलसंवाददाताओं का कहना है कि इस घटना के बाद से पाकिस्तान में भय का ऐसा माहौल पैदा हुआ है कि लोग इस क़ानून का ज़िक्र तक करने से कतराते हैं. इस क़ानून के आलोचकों का कहना है कि इसका इस्तेमाल देश के अल्पसंख्यकों के खिलाफ़ किया गया है और कई बार तो व्यक्तिगत दुश्मनी के मामलों में भी इसका दुरुपयोग होता है. हामन राइट्स वॉच की वरिष्ठ अधिकारी बेडी शेपर्ड का कहना है, "समीउल्लाह के खिलाफ़ एक स्कूल अधिकारी के मामले की शुरुआत किया जाना ही चिंता का विषय है, लेकिन फिर पुलिस और न्यायालय के एक किशोर को जेल भेज देने की घटना आश्चर्यचकित करती है." पुलिस का कहना है कि मोहम्मद समीउल्लाह के खिलाफ़ स्कूल बोर्ड के अधिकारी की शिकायत के बाद केस दर्ज किया गया था. (बीबीसी हिंदी के टर पर भी फ़ॉलो कर सकते हैं.)

Figure 4.1: Dataset Before removing blank spaces and unwanted lines

2188	india	"पिछले हफ्ते रिलीज़ हुई उनकी फ़िल्म 'आर.राजकुमार' को बॉक्स ऑफ़िस पर दर्शकों का अच्छा समर्थन मिला. (कहां घिर गए शाहिद कपूर) फ़िल्म व्यापार विशेषज्ञों के मुताबिक़ फ़िल्म ने पहले सप्ताहांत में करीब 31 करोड़ रुपए का कारोबार किया. उम्मीद की जा रही है कि फ़िल्म पहले सप्ताह में ही अपनी लागत वसूल कर लेगी. शाहिद की इस फ़िल्म को भी समीक्षकों ने बकवास करार दिया था लेकिन दर्शकों ने फ़िल्म को पसंद किया. शाहिद के लिए ये राहत की ख़बर है क्योंकि कुछ दिनों पहले रिलीज़ हुई उनकी फ़िल्म 'फटा पोस्टर निकला हीरो' प्रलोप हो गई थी. बॉलीवुड के 'दबंग' और 'मास्टर ब्लास्टर' सचिन तेंदुलकर एक साथ नज़र आने वाले हैं. किसी फ़िल्म में नहीं बल्कि सेलेब्रिटी क्रिकेट लीग यानी सीसीएल के चौथे संस्करण के लॉन्च के मौक़े पर. इसमें सलमान के छोटे भाई सोहेल की टीम भी शामिल है. ये कार्यक्रम 20 दिसंबर को मुंबई के एक फ़ाइव स्टार होटल में आयोजित होगा. सीसीएल-4 की शुरुआत 25 जनवरी से होगी. इस टूर्नामेंट में विभिन्न फ़िल्मी कलाकारों की टीमों शामिल होंगी. हाल ही में सुप्रीम कोर्ट ने भारत में समलैंगिकता को अपराध घोषित किया है. इस सिलसिले में बॉलीवुड ने समलैंगिकों के पक्ष में आवाज़ उठाई है. सुपरस्टार आमिर ख़ान ने कहा, "मैं बहुत ही निराश हूं. ये फ़ैसला मानवाधिकारों का उल्लंघन है. ये बेहद शर्मनाक बात है." अभिनेता-निर्देशक फ़रहान अख़्तर ने कहा कि सुप्रीम कोर्ट का फ़ैसला ग़लत है. वहीं अभिनेत्री श्रुति हासन ने ट्वीट किया, "ये बात सोच के ही कितनी डरावनी लगती है कि कोई और ये फ़ैसला करे कि हमें किससे प्यार करना चाहिए. यानी अपना साथी चुनने की आज़ादी ही ग़ैरकानूनी घोषित कर दी गई है." करण जोहर और ओनीर ने भी सुप्रीम कोर्ट के फ़ैसले पर निराशा जताई. अभिनेत्री अनुष्का शर्मा ने भी इस फ़ैसले को आज़ादी पर हमला बताया."
2189	international	"सत्रह साल के मोहम्मद समीउल्लाह दक्षिण के शहर कराची में क़ैद हैं. उन पर आरोप है कि उसने एक इम्तिहान के दौरान पेगम्बर मोहम्मद पर अभद्र टिप्पणी की. ह्यूमन राइट्स वॉच ने इस पूरे मामले को 'स्तब्ध करनेवाला' बताया है. पिछले साल नवंबर में एक ईसाई महिला आसिया बीबी को हुई सज़ा के बाद ईश निंदा क़ानून चर्चा में रहा है. हालांकि आसिया बीबी पेगम्बर मोहम्मद के शान में किसी गुस्ताख़ी की बात से इनकार करती हैं. इस साल जनवरी में ही पंजाब के गवर्नर सलमान तासीर की हत्या कर दी गई थी. पुलिस के अनुसार हत्या करनेवाले सलमान तासीर के अंगरक्षक ने कहा कि उसने ऐसा इसीलिए किया क्योंकि तासीर ईश निंदा क़ानून का विरोध कर रहे थे. भय का माहौलसंवाददाताओं का कहना है कि इस घटना के बाद से पाकिस्तान में भय का ऐसा माहौल पैदा हुआ है कि लोग इस क़ानून का ज़िक्र तक करने से कतराते हैं. इस क़ानून के आलोचकों का कहना है कि इसका इस्तेमाल देश के अल्पसंख्यकों के खिलाफ़ किया गया है और कई बार तो व्यक्तिगत दुश्मनी के मामलों में भी इसका दुरुपयोग होता है. हामन राइट्स वॉच की वरिष्ठ अधिकारी बेडी शेपर्ड का कहना है, "समीउल्लाह के खिलाफ़ एक स्कूल अधिकारी के मामले की शुरुआत किया जाना ही चिंता का विषय है, लेकिन फिर पुलिस और न्यायालय के एक किशोर को जेल भेज देने की घटना आश्चर्यचकित करती है." पुलिस का कहना है कि मोहम्मद समीउल्लाह के खिलाफ़ स्कूल बोर्ड के अधिकारी की शिकायत के बाद केस दर्ज किया गया था."
2198	international	"अमरीकी सीनेट कमेटी ने बैंक के बहुत से अधिकारियों से 11 घंटों से भी अधिक समय तक पूछताछ की है. लॉयड ब्लैकफ़्रेन और

Figure 4.2: Dataset Before removing blank spaces and unwanted lines

ii) Dataset Classification: Table shows the categories and subcategories created in the dataset for the classification process. Figure 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.

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

Table 4.1: Summary of Categorization

- Label_0 - International
- Label_1 - India

Figure 4.3: Classification to tag news articles in Hindi



Figure 4.4: Classification to tag news articles in Hindi

iii) Preprocessing:

Machine Learning and NLP techniques can efficiently organize Hindi Newswire articles into various sections. Tokenization, Embedding, stopword removable, Text filtering & cleaning, and vectorization are techniques used in binary classification.

Metrics	Value
Dataset Size	10 MB
Total Number of Documents	5
Total Number of Articles	1825
Avg. Number of Articles per Document	365
Max Number of Article in a Document	777
Min Number of Article in a Document	106
Total Number of Whitespace-Separated Word Tokens in the Raw Dataset	801148
Total Number of Tokens After Pre-Processing and Sentence-Level Tokenization	485968
Token Reduction Rate Post-Preprocessing	39.34%

Table 4.2: Newswire Dataset Representation

4.2.2 HNVec Dataset :

The dataset used in training HNVec Model is called HNVec Dataset and collected from various sources such as www.kaggle.com and github.com. There are a set of structured fields - that include the headlines, full text, authors, timestamps, and labels to train your NLP models and it was collected from various Hindi news-related datasets on Kaggle in both excel and txt format. Initially, all the raw dataset was stored in a single unrefined Excel file with a large size of 1.86 GB.

This raw dataset included:

- Hindi news articles

- Some English text
- Numbers
- URLs (links)
- Irrelevant characters or mixed language content

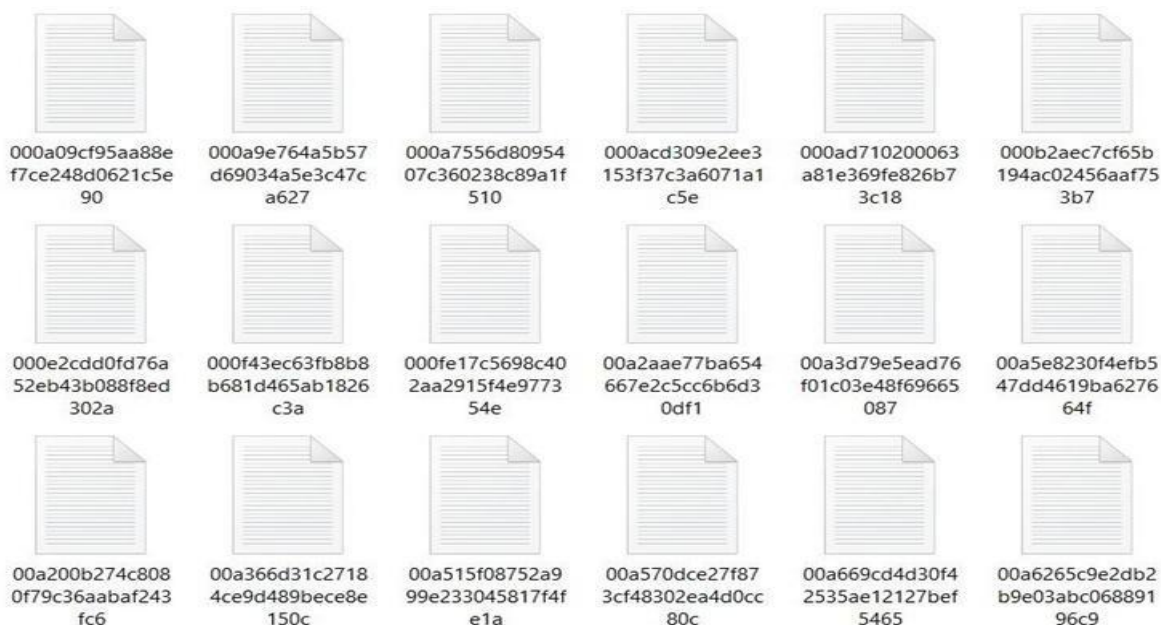


Figure 4.5: Raw Dataset Files

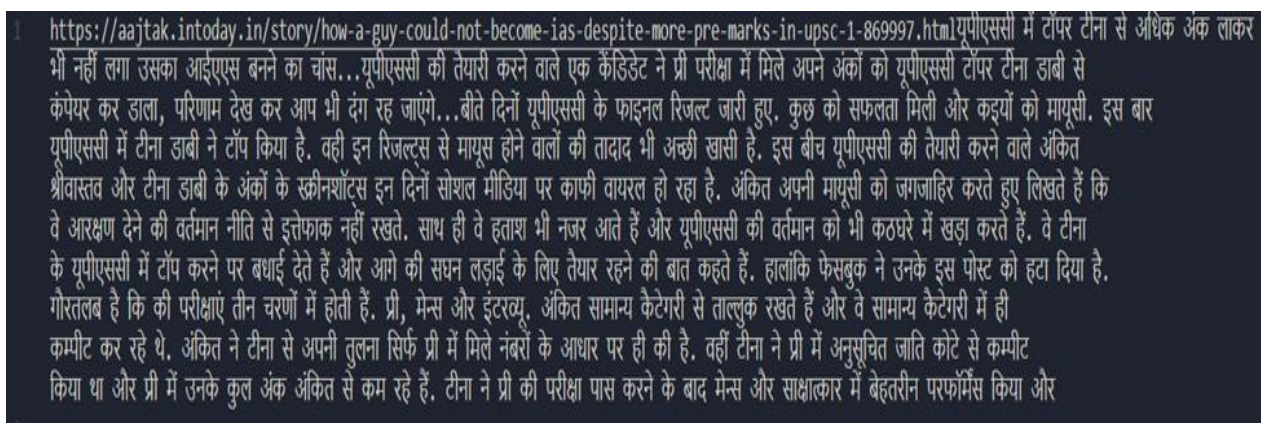


Figure 4.6: Raw Articles

Step 1: Cleaning and Refining the Dataset

To make the data suitable for training HNVec model, a data cleaning process was applied. The goal was to remove anything that could interfere with understanding pure Hindi language text. The cleaning

steps included:

1. Removing English words

English words were removed using regular expressions (`re` module in Python). This helped retain only the Hindi language text, which is essential since the model was trained only on Hindi.

2. Removing numbers

Any digits or numeric values (like dates, statistics, etc.) were removed to ensure that only meaningful language-based features were learned.

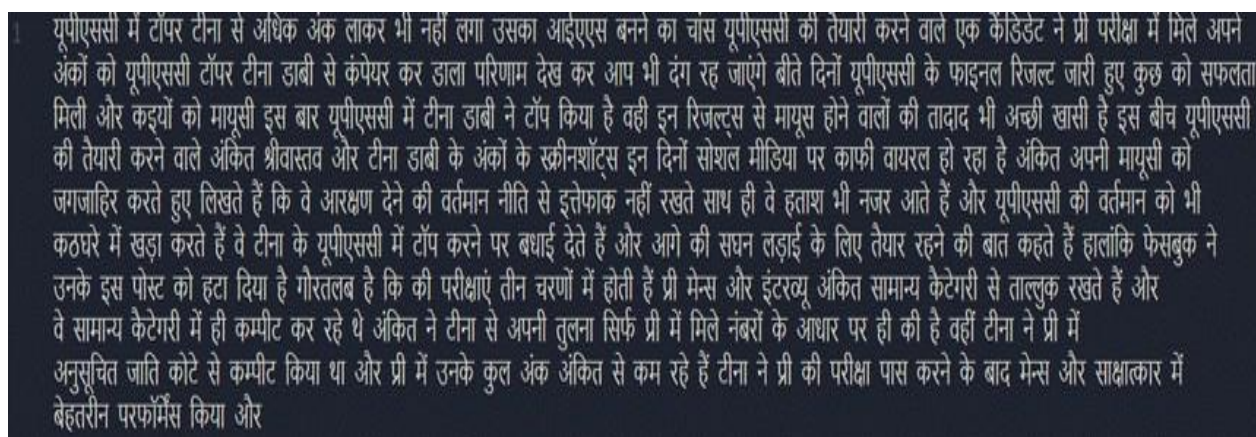
3. Removing hyperlinks/URLs

News articles often contain links, like `http://` or `www.example.com`. These links don't help in understanding the content, so they were removed.

4. Removing special characters

Unnecessary symbols such as `@`, `#`, `*`, `%`, etc., were cleaned from the dataset to avoid noise.

After this process, the dataset was converted into a clean and well-structured CSV file, making it ready for feature extraction and modeling.



यूपीएससी में टॉपर टीना से अधिक अंक लाकर भी नहीं लगा उसका आईएस बनने का चंस यूपीएससी की तैयारी करने वाले एक कैंडिडेट ने प्री परीक्षा में मिले अपने अंकों को यूपीएससी टॉपर टीना डाबी से कंपेयर कर डाला परिणाम देख कर आप भी दंग रह जाएंगे बीते दिनों यूपीएससी के फाइनल रिजल्ट जारी हुए कुछ को सफलता मिली और कइयों को मायूसी इस बार यूपीएससी में टीना डाबी ने टॉप किया है वहीं इन रिजल्ट्स से मायूस होने वालों की तादाद भी अच्छी खासी है इस बीच यूपीएससी की तैयारी करने वाले अंकित श्रीवास्तव और टीना डाबी के अंकों के स्क्रीनशॉट्स इन दिनों सोशल मीडिया पर काफी वायरल हो रहा है अंकित अपनी मायूसी को जगजाहिर करते हुए लिखते हैं कि वे आरक्षण देने की वर्तमान नीति से इत्तेफाक नहीं रखते साथ ही वे हताश भी नजर आते हैं और यूपीएससी की वर्तमान को भी कठघरे में खड़ा करते हैं वे टीना के यूपीएससी में टॉप करने पर बधाई देते हैं और आगे की सघन लड़ाई के लिए तैयार रहने की बात कहते हैं हालांकि फेसबुक ने उनके इस पोस्ट को हटा दिया है गौरतलब है कि की परीक्षाएं तीन चरणों में होती हैं प्री मेन्स और इंटरव्यू अंकित सामान्य कैटेगरी से ताल्लुक रखते हैं और वे सामान्य कैटेगरी में ही कम्पीट कर रहे थे अंकित ने टीना से अपनी तुलना सिर्फ प्री में मिले नंबरों के आधार पर ही की है वहीं टीना ने प्री में अनुसूचित जाति कोटे से कम्पीट किया था और प्री में उनके कुल अंक अंकित से कम रहे हैं टीना ने प्री की परीक्षा पास करने के बाद मेन्स और साक्षात्कार में बेहतरीन परफॉर्मेंस किया और

Figure 4.7 Articles after Cleaning and Refining

Each CSV was loaded using `pandas.read_csv()`, and a **numerical label** was assigned to each category. Then, these labeled datasets were **combined** using `pd.concat()` to form the full dataset.

Step 2: Final Dataset

The final dataset has been arranged into multiple csv files of 1.86 GB. With a total of 700K+ articles.

1	केरल से ऐसी खबर आई जिसने हर किसी को हैरान कर दिया साल के पोमेरियन डॉगी को थिरुवनंतपुरम के छकई के वर्ल्ड मार्केट में
2	बटलर को दिनेश कार्तिक ने जकाती की वाइड गेंद पर स्टम्प आउट किया। वहीं पोलार्ड ने ताम्बे की गेंद पर ड्वेन ब्रावो को सीमारेखा
3	अक्षय कुमार की फिल्म जॉली एलएलबी का पहला गाना गो पागल हाल ही में रिलीज हुआ है यूं तो इंटरनेट पर आते ही इस गाने ने धूम
4	दरअसल सोमवार को कर्नाटक में कई अधिकारियों का तबादला किया गया इनमें हस्सान ज़िले की डिप्टी कमिश्नर रोहिणी सिंदूरी भी हैं
5	अन्य विदेशी मुद्राओं की तुलना में डॉलर की मजबूती और निर्यातकों की ओर से डॉलर की मांग बढ़ने से अंतरबैंक विदेशी मुद्रा बाजार
6	वित्त मंत्री अरुण जेटली ने कहा कि बैंकिंग प्रणाली में फंसे कर्ज की समस्या भारत जैसी बड़ी अर्थव्यवस्था के लिये ऐसी नहीं है जिससे
7	वीरेंद्र सहवाग के कोच एएन शर्मा ने गुरुवार को कहा कि उनके शिष्य साधारण खिलाड़ी नहीं हैं। शर्मा ने आशा जताई कि सहवाग जल्द
8	चीन ने एक बार फिर भारतीय सीमा में घुसपैठ की है। और जुलाई को चीनी सेना के करीब सौ जवान पूर्वी लद्दाख में घुसे। उन जवानों
9	शिवसेना ने मंगलवार को अपने नेता संजय राउत को जून को प्रस्तावित राज्यसभा चुनाव में लगातार तीसरे कार्यकाल के लिए उम्मीद
10	प्रिंस ऑफ कोलकाता और भारतीय क्रिकेट टीम के पूर्व कप्तान सौरव गांगुली तथा क्रिकेट के सभी संस्करणों में विस्फोटक बल्लेबाजी
11	रेलवे इस साल लाखों पदों पर भर्तियां करने वाला है इनमें आरआरबी एनटीपीसी ग्रुप डी पैरामेडिकल और मिनिस्ट्रियल और आइ
12	बॉलीवुड एक्ट्रेस आलिया भट्ट और एक्टर रणबीर कपूर को लेकर आए दिन कोई न कोई खबर मीडिया के सुर्खियों में बनी रहती है
13	हेडलाइन के अलावा इस खबर को एनडीटीवी टीम ने संपादित नहीं किया है यह सिंडीकेट फीड से सीधे प्रकाशित की गई है।
14	इंग्लैंड के एलेक्स ट्यूडर तीसरे ऐसे बल्लेबाज बने जो नाबाद होने के बावजूद सिर्फ एक रन से शतक पूरा करने से चूक गए ट्यूडर व
15	सीमा पूनिया ने इस सत्र का अपना सर्वश्रेष्ठ प्रदर्शन करते हुए राष्ट्रमंडल खेलों में महिलाओं के चक्का फेंक में रजत पदक जीता लेकिन
16	भारतीय ओलिंपिक संघ ने रियो के लिए खिलाड़ियों की विदाई के मौके पर सितारों की महफ़िल भी सजा दी। सलमान खान और एआ
17	अजिंक्य रहाणे की एक ओवर में छह चौकों के रिकार्ड प्रदर्शन से सजी नाबाद शतकीय पारी और बोल्ड के मास्टर बने सिद्धार्थ त्रिवेदी ने
18	उत्तर प्रदेश के कानपुर में एक बड़ा ट्रेन हादसा हुआ है कानपुर में हावड़ा से दिल्ली आने वाली पूर्वा एक्सप्रेस बेपटरी हो गई है बताया
19	बता दें रेलवे समय सारिणी रेल मंत्रालय द्वारा नियमित इंटरसिटी और लंबी दूरी की ट्रेन यात्रियों के साथसाथ विदेशी और घरेलू पर्यटकों

Figure 4.8: Final Dataset CSV file

Problems arose during creation of Dataset:

Working with the NLP dataset poses significant memory challenges. For instance a dataset of 50 MB with a vocabulary size of 56,000 required approximately 12.5 GB of GPU RAM to compute the co-occurrence matrix. The memory usage arises from the temporary structures created during tokenization, vectorization, and embedding processes.

When scaled to the largest dataset, the memory requirement becomes massive. A 1.86 GB dataset with a vocabulary of 550,000 would produce a co-occurrence matrix with over 302 billion entries, each requiring 4 bytes, totaling around 1.1 TB of memory. Even processing a 500 MB dataset with a vocab size over 200,000 would need approximately 150 GB of GPU RAM.

To handle this issue, the input size was limited to smaller chunks like 50 MB to manage GPU memory more efficiently. This highlighted the importance of optimizing memory usage in deep learning applications and avoiding wasteful resource consumption, especially when working with NLP models and large scale datasets.

Metric	Value
Dataset Size	50 Mb
Total Number of Documents	1
Total Number of Articles	56949
Total Number of Whitespace-Separated Word Tokens in the Raw Dataset	3939982
Total Number of Tokens After Pre-Processing and Sentence-Level Tokenization	2535219
Token Reduction Rate Post-Preprocessing	35.65%
Vocabulary Size of the Pre-Processed Corpus	56123

Table 4.3: HNVec Dataset Representation

4.3 Hybrid Classifier Models integrated with vectorizer

The method implemented is to create a scalable and effective hybrid learning framework that combines deep learning and conventional machine learning models with a unique vectorization technique along with the key idea of leveraging the feature extraction capabilities of CNN and combine them with the classification strengths of both Logistic Regression (LR) and Support Vector Machine (SVM). CNN is used to automatically learn and extract meaningful features from the data, while LR and SVM are applied for the final classification based on these features. This hybrid approach integrates deep learning and traditional machine learning techniques to create a model that can capture complex patterns while maintaining robust classification performance.

Three different vectorization techniques i.e. a tailored context dependent word embedding vectorizer (HNVec), Term Frequency-Inverse Document Frequency (TF-IDF), FastText embeddings are used to systematically test the hybrid CNN-LR-SVM architecture in order to examine the effects of various textual representations on model performance. Before being fed into the CNN module to extract high-level abstract characteristics, each vectorizer converts the preprocessed text into numerical representations. After that, both LR and SVM classifiers receive these features for comparison. We

can choose the best representation technique for Hindi text categorization within the suggested hybrid learning framework thanks to the multi-vectorizer strategy, which permits a thorough performance evaluation.

4.3.1 HLM-CLS with HNVec

HNVec is an organized method for producing excellent vector embeddings for Hindi news items. To enhance word representations, HNVec uses a home-grown preprocessing pipeline in conjunction with a distance-based co-occurrence matrix technique. The drawbacks of the existing English and multilingual embeddings, which frequently overlook the linguistic richness of Hindi, are addressed by the Hindi-specific vectorization model HNVec.

We present a hybrid CNN-LR-SVM architecture that integrates context-dependent word embeddings with a tailored vectorizer based on a distance-weighted co-occurrence matrix to deal with the problem of accurate Hindi news classification. The vectorization technique, which employs a distance-based co-occurrence matrix to capture the frequency and positional proximity of word pairs within a specified sliding window, is applied after raw Hindi news articles have first undergone dataset processing. A Convolutional Neural Network (CNN) processes the resultant input, after which LR and SVM are applied. By combining precise vocabulary mapping, high-dimensional embeddings, and robust preprocessing, the HNVec GloVe model efficiently processes and categorizes Hindi news.

4.3.2 HLM-CLS with TF-IDF

This is a Hybrid Learning Algorithm for Text classification using a triplet combination of CNN- LR-SVM along with TF-IDF. This concept's core idea is to determine the significance of each word and document the connections between them. This model proves especially useful when deep learning techniques alone are not sufficient for classification refinement or when traditional models have difficulty to capture intricate patterns from high dimensional news dataset.

The HLM-CLS model combines the approach of extracting deep relationships between words and classification of key feature weights, utilizing feature extraction through CNN, followed by classification with LR and SVM, often incorporating TF-IDF vectorization for additional feature representation. It is observed that this classifier showed moderate performance as the vectorizer is not working effectively on the Hindi news dataset. TF-IDF is causing sparsity issues, lack of context understanding and Mismatch in Vocabulary. The model achieves average news classification accuracy

and lacks in predicting some of the classes. It may be particularly suitable for applications that require both deep word-level feature extraction and sophisticated classification if the sparsity issue is resolved with use of advanced word embedding methods.

4.3.3 HLM-CLS with FastText

In this work, we provide a hybrid architecture for classifying Hindi news articles into various specified categories via incorporating CNN-LR-SVM modeling with FastText word embeddings. The pipeline starts with preparatory operations including normalization, tokenization, stopword removal, and lemmatization that are specific to Hindi. The FastText vectorizer, which offers context-aware word embeddings that take subword information into account, is then used to convert each tokenized document. This is especially helpful for morphologically rich languages like Hindi, where lexical coverage and semantic understanding are improved by FastText's capacity to create embeddings for uncommon and out-of-vocabulary terms. Upon being reshaped, the dense vector representations of documents are input into a Convolutional Neural Network (CNN), which uses max-pooling and convolution to identify hierarchical features and local patterns. For the final prediction, these deep feature representations are then flattened and fed into two parallel classifiers: Support Vector Machine (SVM) and Logistic Regression (LR). Comparing probabilistic and margin-based decision boundaries over the same feature space is made possible by this dual-classifier method. The effectiveness of combining CNN-driven feature extraction, classical classifiers, and FastText embeddings for robust multilingual text classification. However, FastText's reliance on local subword-level information may cause it to perform poorly when capturing long-range dependencies and global semantic links, even while it provides high generalization and robustness for morphologically complex tokens.

4.4 HNVec: A Novel Vectorizer for Hindi News Classification

HNVec (Hindi News Vectorizer) adopts a GloVe-style learning objective and distance-based weighting for co-occurrence statistics to create context-aware embeddings. It integrates CNN- LR-SVM into a single pipeline for text categorization and leverages the strengths of both deep learning and classical machine learning models. The primary aim of this hybrid approach is to handle complex decision boundaries, effectively categorize text, and capture local features. It is particularly suited for applications that require both detailed feature extraction and sophisticated classification. This model proves especially useful when deep learning techniques alone are insufficient for classification refinement or when traditional models struggle to capture intricate patterns. The following section

presents an expression for the distance-based co-occurrence matrix that is employed for the suggested method's HNVec.

4.4.1 Co-occurrence Matrix

A co-occurrence matrix is a mathematical representation that captures the frequency with which pairs of words appear together within a specified context, such as a sentence, paragraph, or document. It is a square matrix where rows and columns represent unique words in the corpus, and each cell (I, j) contains the number of times word I appears in the context of word j.

1. Co-occurrence Weighting Formula:

$$cooc(w_i, w_j) = \frac{1}{|i - j|}$$

where:

- 1) i is the target word at position in a sentence.
- 2) j is a context word within a window .
- 3) $|i - j|$ is the absolute distance between the two positions.

2. Window Boundaries Calculation:

Window boundaries impact the creation of co-occurrence matrices in embedding models by determining the span of context words surrounding a target word. Because they restrict the range of context, they are essential for capturing syntactic and semantic links.

$$\begin{aligned} start &= \max(0, i - window_size) \\ end &= \min(L, i + window_size + 1) \end{aligned}$$

where,

- 1) i is the index of the target word.
- 2) L is the length of the sentence.
- 3)

3. Co-occurrence Matrix Construction:

One of the initial steps in natural language processing (NLP) is creating a co-occurrence matrix, which identifies word associations based on context-based proximity. Many word embedding techniques eventually rely on a co-occurrence matrix to transform the text into

meaningful numbers.

$$C[\mathbf{w}_i][\mathbf{w}_j] = \sum_{occurrences} \frac{1}{|i - j|}$$

where,

- 1) $C \in \mathbb{R}^{V \times V}$ and V is the total vocabulary size.
- 2) \mathbf{w}_i is the target word at position i in a sentence.
- 3) \mathbf{w}_j is a context word within a window around \mathbf{w}_i .
- 4) $|i - j|$ is the absolute distance between the two positions.

4.4.2 HNVec Model Training:

In order to develop context-sensitive representations of Hindi words, the HNVec (Hindi News Vectorizer) model is a customized word embedding generator that draws inspiration from GloVe. Using a distance-based co-occurrence matrix, it combines local syntactic closeness with global statistical co-occurrence data.

1. Weighting Function:

$$f(x) = \begin{cases} \left(\frac{x}{x_{max}}\right)^\alpha & \text{if } x < x_{max} \\ 1 & \text{otherwise} \end{cases}$$

Where:

- 1) x is Co-occurrence count between word i and context word j .
 - 2) x_{max} is Cutoff threshold for weighting.
 - 3) α is Smoothing exponent for weighting function.
2. Dot Product of Word and Context Embeddings:

$$\text{dot}_{ij} = W_i \cdot W_j^{\text{context}} = \sum_{k=1}^d W_{ik} \cdot W_{jk}^{\text{context}}$$

Where:

- 1) W_i : Embedding vector for word i , of size d .
- 2) W_j^{context} : Context embedding vector for word j , of size d .
- 3) d : Dimensionality of embeddings
- 4) \cdot : Dot product between vectors

3. Cost Function:

$$\text{cost}_{ij} = W_i \cdot W_j^{\text{context}} + b_i + b_j^{\text{context}} - \log(X_{ij})$$

Where

- 1) b_i : bias for word i .
- 2) b_j^{context} : Bias for context word j .
- 3) $\log(X_{ij})$: Natural logarithm of co-occurrence count between words i and j

4. Weighted Cost:

$$\text{weighted_cost}_{ij} = f(X_{ij}) \cdot \text{cost}_{ij}$$

Where:

- 1) $f(X_{ij})$: Weighting function applied to co-occurrence value.
- 2) cost_{ij} : Raw error for the word pair (i, j) .

5. Lost Function:

$$\mathcal{L} = \frac{1}{2} \sum_{(i,j)} f(X_{ij}) \cdot (W_i \cdot W_j^{\text{context}} + b_i + b_j^{\text{context}} - \log(X_{ij}))^2$$

Where:

1) \mathcal{L} : Overall loss.

- Word embedding gradient:

$$\nabla W_i = f(X_{ij}) \cdot \text{cost}_{ij} \cdot W_j^{\text{context}}$$

- Context embedding gradient:

$$\nabla W_j^{\text{context}} = f(X_{ij}) \cdot \text{cost}_{ij} \cdot W_i$$

- Bias gradient (word and context):

$$\nabla b_i = f(X_{ij}) \cdot \text{cost}_{ij}$$

$$\nabla b_j^{\text{context}} = f(X_{ij}) \cdot \text{cost}_{ij}$$

2) $\sum_{(ij)}$: Sum over all word-context pairs.

6. Gradients:

Where:

1) η : Learning rate.

2) 0.1 : Scaling factor applied to bias updates.

7. Parameter Updates:

$$W_i \leftarrow W_i - \eta \cdot \nabla W_i$$

$$W_j^{\text{context}} \leftarrow W_j^{\text{context}} - \eta \cdot \nabla W_j^{\text{context}}$$

$$b_i \leftarrow b_i - \eta \cdot 0.1 \cdot \nabla b_i$$

$$b_j^{\text{context}} \leftarrow b_j^{\text{context}} - \eta \cdot 0.1 \cdot \nabla b_j^{\text{context}}$$

Where:

1) η : Learning rate.

2) 0.1 : Scaling factor applied to bias updates

4.4.3 Algorithm for HNVec with HLM-CLS

HLM-CLS: Hybrid Learning Model using CNN-LR-SVM	
<u>Input Processing:</u>	
<ol style="list-style-type: none"> 1. Import all the libraries. 2. Load the news datasets and convert them into pandas DataFrames. 3. Assign labels to each news category. 4. Combine all the category DataFrames into a single DataFrame. 5. Define features and labels such that: $X = \text{news['news_articles']}$ $y = \text{news['label']}$ 6. Create a list of stop words and define a tokenizer that divides text into tokens. 	
<u>Feature Extraction:</u>	
<ol style="list-style-type: none"> 7. Following are the steps for Feature Extraction by using trained HNVec Embeddings: <ul style="list-style-type: none"> - Load the pre-trained HNVec embeddings stored in a pickle file. - Extract the word-to-index mapping and the embedding matrix from the pickel file. - For each document within the dataset: <ul style="list-style-type: none"> - Tokenize the document with whitespace. - For every token: <p>If the token exists in the vocabulary (word_to_id), retrieve the corresponding vector.</p> - Calculate the average of all token embeddings to create a fixed-length document embedding. - If no valid tokens exist in the document, return a zero vector. - Generate <i>HNVec Features</i> for train and test split. 	
<u>Mathematical Representation:</u>	
<p>Given:</p> $Di = \text{document } i$ $wj \in Di = \text{tokens in the document}$ $E(wj) \in \mathbb{R} = \text{word embedding vector of } wj$ $N = \text{number of valid tokens in the document found in the HNVec vocabulary.}$ <p>Then:</p>	

$$HNVec(Di) = \begin{cases} \frac{1}{N} \sum_{j=1}^N E(w_j) & \text{if } \exists w_j \in vocab \\ 0 & \text{otherwise} \end{cases}$$

8. Steps for CNN Model:

- Convert the text data into integers using tokenization.
- Pad the converted data.
- Initialize a CNN model with three components: global max-pooling, convolution, and embedding.
- Train the CNN model by using the training data.

The Convolution Operation of the CNN model considering an input sequence `X` of length `n`, and a filter `w` of size `k` is defined as:

$$\text{Convolution Output} = X * w = \sum_{i=1 \text{ to } k} X[i] \times w[i]$$

The Global Max-Pooling operation of the CNN model considering the convolution output `Z` is defined as:

$$\text{Global Max - Pooling}(Z) = \max(Z)$$

- Use the trained CNN model to extract features from the training and testing datasets.

$$CNN \text{ Features} = CNN(X_train_pad)$$

Feature Combination:

9. Combine HNVec and CNN Features:

$$\text{Train Features} = HNVec \text{ Features} \oplus CNN \text{ Features}$$

$$\text{Test Features} = HNVec \text{ Features} \oplus CNN \text{ Features}$$

where \oplus denotes concatenation

Feature Reduction:

10. Initialize an SVM model and a Logistic Regression model.

11. Use the HNVec and CNN combined features to train the SVM and Logistic Regression model.

If `w` is weight vector, and `b` is bias.

For SVM is defined using the Decision Function given as:

$$f(X) = \text{sign}(w \times X + b)$$

To predict Logistic Regression Probability Sigmoid function is defined as:

$$P(y = 1|X) = \sigma(w \times X + b) = 1 / (1 + e^{-(w \times X + b)})$$

12. Using trained SVM and Logistic Regression models, predict probabilities for both training and testing sets.

13. Combine both SVM and Logistic Regression Probabilities to get reduced feature set:

$$Y_combined = SVM \text{ Probabilities} \oplus \text{Logistic Regression Probabilities}$$

Final Model and Evaluation:

14. By using the reduced feature set train a Final Support Vector Machine (SVM) model.
15. Generate predictions for both the training and test data by using the final SVM model.

$$\hat{Y} = f(Y_combined)$$

16. Measure the performance using the following metrics:

$$Accuracy = \frac{(Number\ of\ correct\ predictions)}{(Total\ number\ of\ predictions)}$$

$$Recall = \frac{(True\ Positives)}{(True\ Positives + False\ Negatives)}$$

$$Precision = \frac{(True\ Positives)}{(True\ Positives + False\ Positives)}$$

$$F1 - Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$$

- Compute the training and validation loss history from the CNN model training.

17. Output Results.

4.4.4 Pseudo code for HNVec

Vocabulary Mapping:

```
DEFINE word_counts:  
    Count each unique word in a sentence of tokenized_sentences  
DEFINE vocab:  
    Make a list of unique words in  
word_counts DEFINE vocab_size  
DEFINE word_to_id, id_to_word;
```

Compute Co-occurrence Matrix:

```
FUNCTION COMPUTE_COOCCURRENCE_GPU(token_ids):  
    FOR sentence IN token_ids:  
        FOR i IN RANGE(LENGTH(sentence)):  
            word_id = sentence[i]  
            FOR j IN RANGE(MAX(0, i - window_size), MIN(LENGTH(sentence), i + window_size + 1)):  
                IF i ≠ j:  
                    cooccurrence_matrix[word_id][sentence[j]] += 1.0 / ABS(j - i)  
  
            IF INDEX(sentence) MOD 5000 = 0:  
                PRINT("Processed", INDEX(sentence), "sentences...")  
  
    SAVE_SPARSE_MATRIX(CONVERT_TO_SPARSE_COO(TRANSFER_TO_CPU(cooccurrence_matrix)), save_path)  
    PRINT("Final matrix saved at:", save_path)  
END FUNCTION
```

Load Sparse Matrix for Training:

```
LOAD the sparse co-occurrence matrix from the saved file  
  
CONVERT the matrix to COO (Coordinate List) format, if  
necessary INITIALIZE an empty dictionary cooccurrence  
  
FOR each non-zero entry in the sparse  
matrix: EXTRACT row index  
      (word_id) EXTRACT column index  
      (context_id) EXTRACT co-occurrence  
      count  
      STORE in dictionary as key-value pair {(word_id, context_id): co-occurrence_count}
```

HNVec Model Initialization:

```
DEFINE device AS "cuda" IF CUDA is available ELSE "cpu"
DEFINE FUNCTION weighting(x)
  RETURN (x / x_max) ^ alpha IF x <
    x_max ELSE 1
END FUNCTION
DEFINE vocab_size AS
LENGTH(word_to_id)
DEFINE model
parameters
  SET W, W_context AS random tensors (vocab_size, embedding_dim) - 0.5
  SET bias, bias_context AS zero tensors (vocab_size)
  MOVE W, W_context, bias, bias_context TO
device
END DEFINE
DEFINE co-occurrence data processing
  SET pairs, values AS KEYS, VALUES FROM cooccurrence dictionary
  CONVERT pairs TO tensor (dtype=long), values TO tensor
  (dtype=float32)
  MOVE pairs_tensor, Xij_tensor TO device
  COMPUTE log_Xij_tensor AS LOG(Xij_tensor)
END DEFINE
DEFINE batch_size AS 100000
```

Train HNVec Model:

```
DEFINE FUNCTION train_model(epochs, num_pairs, pairs_tensor, Xij_tensor, log_Xij_tensor, batch_size, device, lr, W,
W_context, bias, bias_context):
  FOR epoch FROM 0 TO epochs - 1:
    total_loss = 0.0
    perm = random_permutation(num_pairs, device)
    pairs, Xij, log_Xij = pairs_tensor[perm], Xij_tensor[perm], log_Xij_tensor[perm]

    FOR start FROM 0 TO num_pairs STEP batch_size:
      i, j = extract_first_column(pairs[start TO start+batch_size]), extract_second_column(pairs[start TO
start+batch_size])
      cost = sum(W[i] * W_context[j], axis=1) + bias[i] + bias_context[j] - log_Xij[start TO
start+batch_size]
      weight = weighting(Xij[start TO
start+batch_size])
      loss = 0.5 * sum(weight *
cost**2)
      total_loss += loss

      grad_w = expand_dims(weight *
cost)
      W[i] -= lr * grad_w *
W_context[j]
      W_context[j] -= lr *
grad_w * W[i]
      bias[i] -= lr * 0.1 *
weight * cost
      bias_context[j] -= lr *
0.1 * weight * cost

    PRINT "Epoch", epoch + 1, "/", epochs, "Loss:", total_loss
  END FUNCTION
```

Save Final Word Embeddings:

```
DEFINE FUNCTION save_embeddings(W, W_context, file_path)
  COMBINE embeddings
  SAVE to
file END
FUNCTION
```

Save Model in Pickle Format:

```
DEFINE FUNCTION save_model(word_to_id, id_to_word, embeddings, file_path)
  STORE mappings and embeddings in a structured format
  SAVE to
file END
FUNCTION
```


4.4.5 Flow Chart

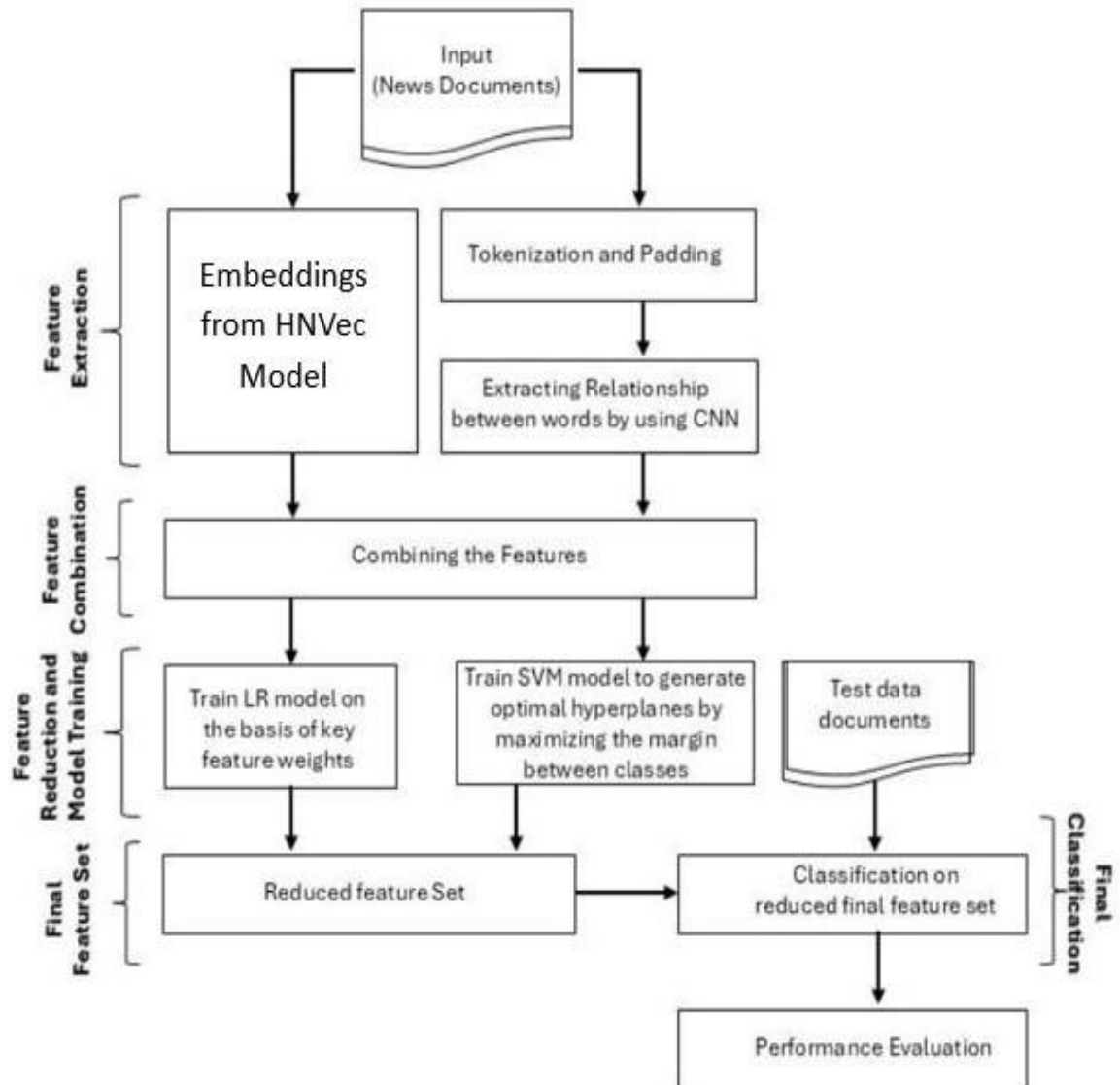


Figure 4.9: Flow Chart

4.4.6 Architecture

The CNN-LR-SVM model shown in the figure represents an advanced strategy that integrates both machine learning and deep learning for text categorization, utilizing feature extraction through CNN, followed by classification with LR and SVM, often incorporating HNVec vectorization for additional feature representation.

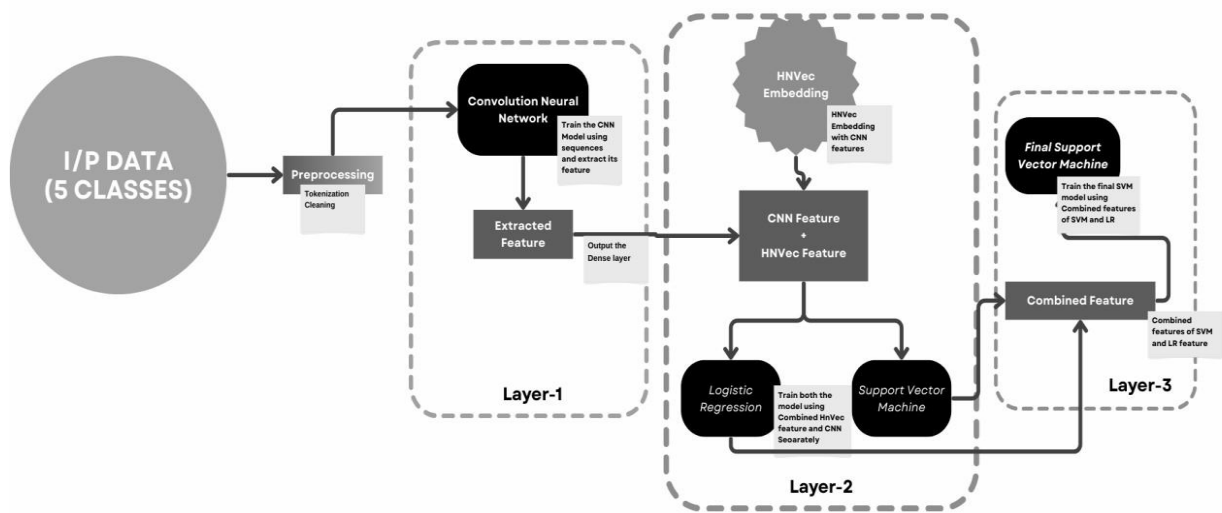


Figure 4.10: Architecture of HNVec with HLM-CLS

4.5 SUMMARY

Chapter 4, Methods and materials we discuss the methods implemented to carry out the various activities in the dataset cleaning and preprocessing stage and analyze the machine learning model for classifying Hindi newswire articles. In this chapter, in section 4.2 we discuss about dataset preparation which includes text preprocessing and feature extraction, in section 4.3 implementation of various hybrid classifier models integrated with vectorizer are illustrated, in section 4.4 we discuss about noble vectorizer HNVec along with its algorithm and flow diagram.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 INTRODUCTION

In any research work, results and discussions are very crucial. In chapter 4 we already discuss the implementation methods of various three-layer models incorporated with different embeddings. In this chapter, we are going to evaluate the performance of those vectorizers and to analyze a well-structured hybrid approach that appears to be best, considering its high accuracy and less losses in terms of accuracy, precision, F1 score, and recall, achieving a better accuracy rate. The results also highlight that consistency between the size of the training set, choice of vectorizer, the number of classes, and effective data preparation are crucial factors influencing the technique's effectiveness.

5.2 VECTORIZER ANALYSIS

The balance between computing efficiency, accuracy, simplicity, and informativeness is taken into consideration as several hybrid classifier models using different embedding techniques are examined and contrasted using important metrics such as accuracy, F1 score, recall, and precision. The ultimate goal was to use these factors to determine and suggest the combinations that performed the best.

5.2.1 Performance analysis of HLM-CLS with HNVec

HNVec (Hindi News Vectorizer), context dependent word embedding utilizing distance-based co-occurrence matrix is incorporated with HLM-CLS (Hybrid Learning Model- CNN-LR-SVM) is an efficient technique for Hindi News Classification. The output table below shows all the important matrices.

Model	Train: Test	Epoch	Training Accuracy	Testing Accuracy	Precision	Recall	F1 Score	Validation Loss	Training Loss
HNVec	90:10	10	94.23	85.29	85.47	85.29	85.13	50.10	47.00
		20	94.17	85.29	85.47	85.29	85.13	50.10	49.71
		30	96.40	87.06	87.30	87.60	87.02	50.10	25.64
		40	96.53	85.88	86.49	85.88	85.90	46.95	24.16

		50	96.53	85.29	85.53	85.29	85.26	43.50	24.69
	80:20	10	98.01	87.65	88.32	87.65	86.79	64.31	20.59
		20	95.87	87.06	86.99	87.06	86.87	39.61	20.84
		30	94.40	86.47	86.27	86.47	86.28	50.10	49.88
		40	94.40	86.47	86.27	86.47	86.28	50.10	49.95
		50	94.47	86.76	86.54	86.76	86.54	50.10	49.44
	70:30	10	94.52	88.61	88.50	88.61	88.51	51.81	50.31
		20	94.36	88.61	88.50	88.61	88.51	50.10	49.97
		30	94.36	88.61	88.50	88.61	88.51	50.10	49.79
		40	94.36	88.61	88.50	88.61	88.51	50.10	49.93
		50	94.36	88.61	88.50	88.61	88.51	50.21	50.02

Table 5.1: Performance Analysis Table for HLM-CLS with HNVec

Above table indicates that using HNVec along with CNN-LR-SVM model is the most effective model since it balances Recall, Precision, and F1 Score while achieving the best test accuracy.

The performance of a classification model trained for Hindi news classification is displayed in the confusion matrix. The training and validation loss graph shows the following patterns when a Hindi news classification model is being trained using a **70-30 train-test split** and assessed after 10 epochs, 20 epochs, 30 epochs, 40 epochs and 50 epochs. In the confusion matrix, each row represents the actual class, and each column represents the predicted class.

The following figures represent Confusion Matrix, Training and Validation Loss graphs for 70:30 Train:Test split with 30 epoch:

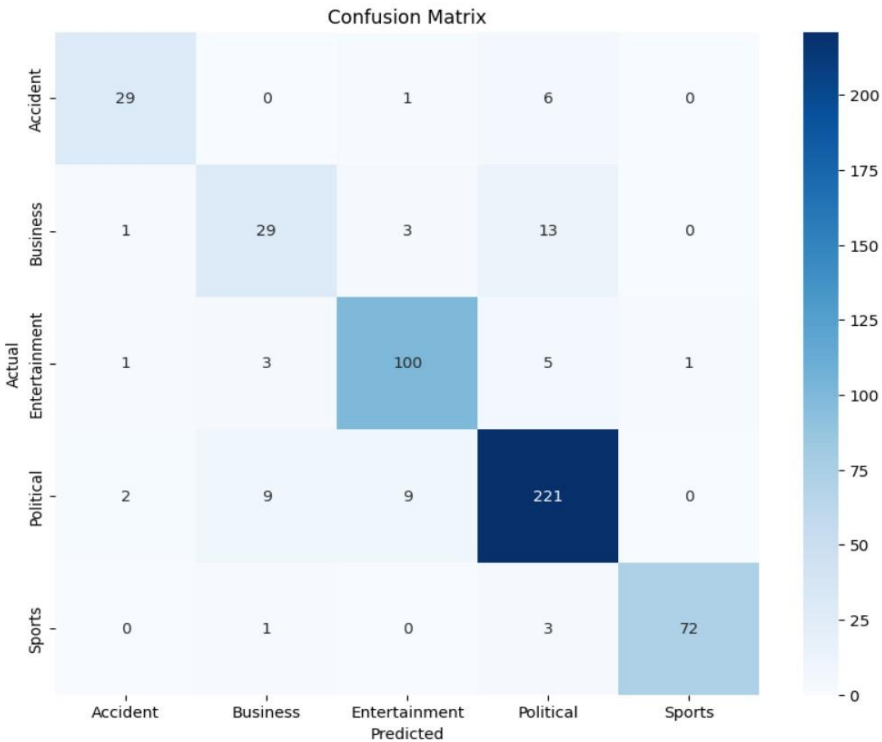


Figure 5.1: Confusion Matrix of HLM-CLS with HNVec for 30 Epoch

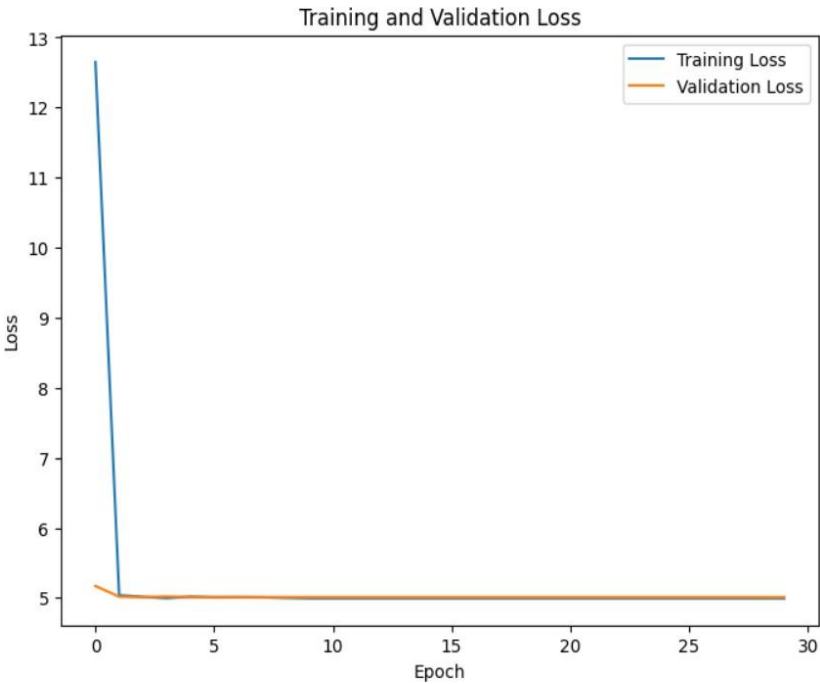


Figure 5.2: Training vs Validation Loss of HLM-CLS with HNVec for 30 Epoch

The above shown Figure 5.1 Confusion Matrix suggests that the model performs well overall. The model has high accuracy in predicting politics for 30 epochs including 221 correct prediction and good performance in entertainment. However, the model struggles with the misclassification between business and politics categories.

The above plot Figure 5.2 demonstrates how the validation loss of HNVec embeddings stays continuously low while the training loss rapidly drops and stabilizes within the first few epochs. This suggests that the HNVec-trained model converges rapidly and has good generalization without overfitting.

5.2.2 Performance analysis of HLM-CLS with TF-IDF

Combining CNN features with TF-IDF features provides a more comprehensive representation of the textual data by capturing both shallow statistical relationships (TF-IDF) and deep semantic patterns (CNN). The output table below shows all of the important matrices.

Model	Train: Test	Epoch	Training Accuracy	Testing Accuracy	Precision	Recall	F1 Score	Validation Loss	Training Loss
TF-IDF	90:10	10	94.23	76.47	76.44	76.47	76.77	3.19	0.53
		20	92.73	77.06	77.99	77.06	76.92	4.07	2.53
		30	88.79	77.06	77.23	77.06	76.97	4.65	4.57
		40	97.58	64.71	69.36	64.71	60.22	6.06	1.47
		50	99.48	71.18	72.19	71.18	66.03	3.82	1.06
	80:20	10	99.41	75.00	75.21	75.00	72.78	3.72	0.31

		20	95.80	70.29	72.25	70.29	67.70	5.65	3.09
		30	89.53	80.88	80.54	80.88	80.49	4.60	4.47
		40	94.91	76.76	79.08	76.76	77.03	4.52	1.41
		50	89.38	81.18	81.03	81.18	80.97	4.60	4.55
	70:30	10	89.72	81.14	81.45	81.14	81.28	4.41	4.00
		20	89.13	81.34	81.45	81.34	81.39	4.60	4.47
		30	89.05	81.14	81.33	81.14	81.23	4.61	4.59
		40	99.16	67.58	78.29	67.58	70.29	2.86	0.64
		50	89.13	81.34	81.47	81.34	81.39	4.60	4.58

Table 5.2: Performance Analysis Table for HLM-CLS with TF-IDF

The performance of a classification model trained for Hindi news classification is displayed in the confusion matrix. The training and validation loss graph shows the following patterns when a Hindi news classification model is being trained using a **70-30 train-test split** and assessed after 10 epochs, 20 epochs, 30 epochs, 40 epochs and 50 epochs. In the confusion matrix, each row represents the actual class, and each column represents the predicted class.

The following figures represent Confusion Matrix, Training and Validation Loss graphs for 70:30

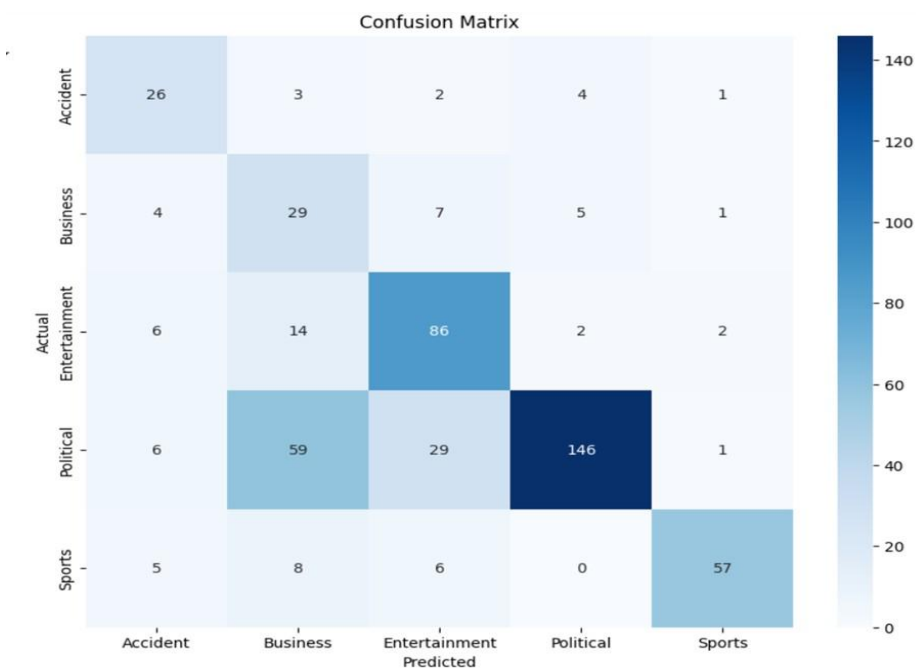


Figure 5.3: Confusion Matrix of HLM-CLS with TF-IDF for 40 Epoch

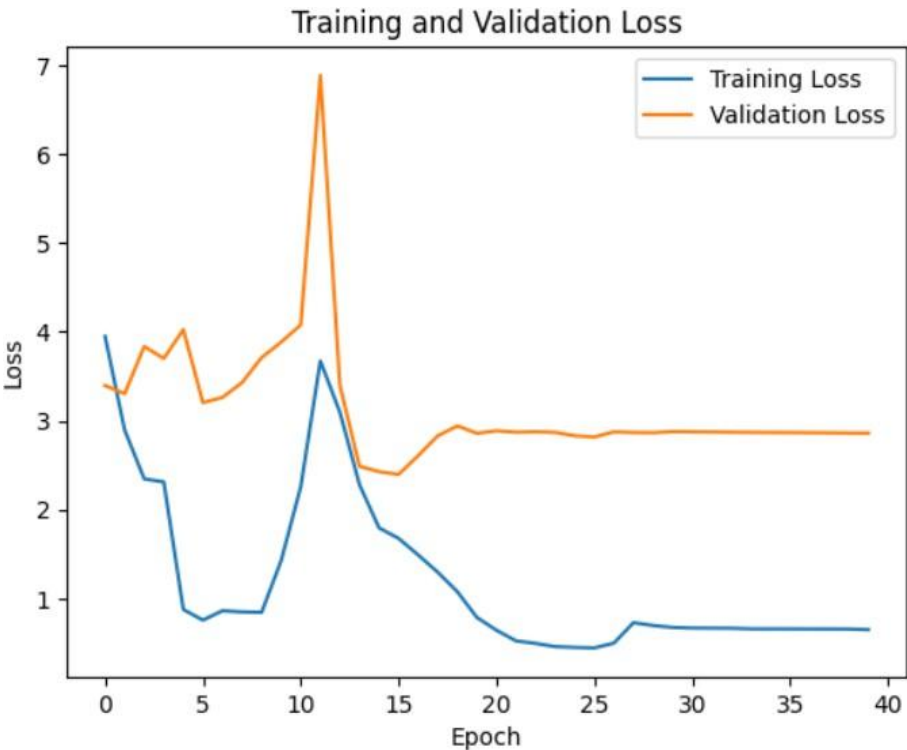


Figure 5.4: Training vs Validation Loss of HLM-CLS with TF-IDF for 40 Epoch

The above shown Figure 5.3 Confusion Matrix suggests that the model performs well overall. The model has high accuracy in predicting politics for 40 epochs including 146 correct prediction and good performance in entertainment. However, the model struggles with the misclassification between business and politics categories.

The above plot Figure 5.4 illustrates high variations in the TF-IDF training and validation loss curves suggest instability during training. Overfitting and inadequate generalization are suggested by the validation loss, which is still much more than the training loss.

5.2.3 Performance analysis of HLM-CLS with FastText

The output table below shows all of the important matrices.

Epoch	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score	Training Loss	Validation Loss
10	96.01	75.88	76.46	75.88	74.47	0.03	0.74
20	96.54	76.18	76.92	76.18	75.02	0.02	0.83
30	97.86	77.35	77.58	77.35	75.75	0.00	0.88
40	98.19	80.00	80.36	80.00	78.79	0.00	0.92
50	97.85	77.65	77.73	77.65	76.39	0.00	0.97

Table 5.3: Performance Analysis Table for HLM-CLS with FastText

The performance of a classification model trained for Hindi news classification is displayed in the confusion matrix. The training and validation loss graph shows the following patterns when a Hindi news classification model is being trained using a **70-30 train-test split** and assessed after 10 epochs, 20 epochs, 30 epochs, 40 epochs and 50 epochs. In the confusion matrix, each row represents the actual class, and each column represents the predicted class.

The following figures represent Confusion Matrix, Training and Validation Loss graphs for 30 epoch:

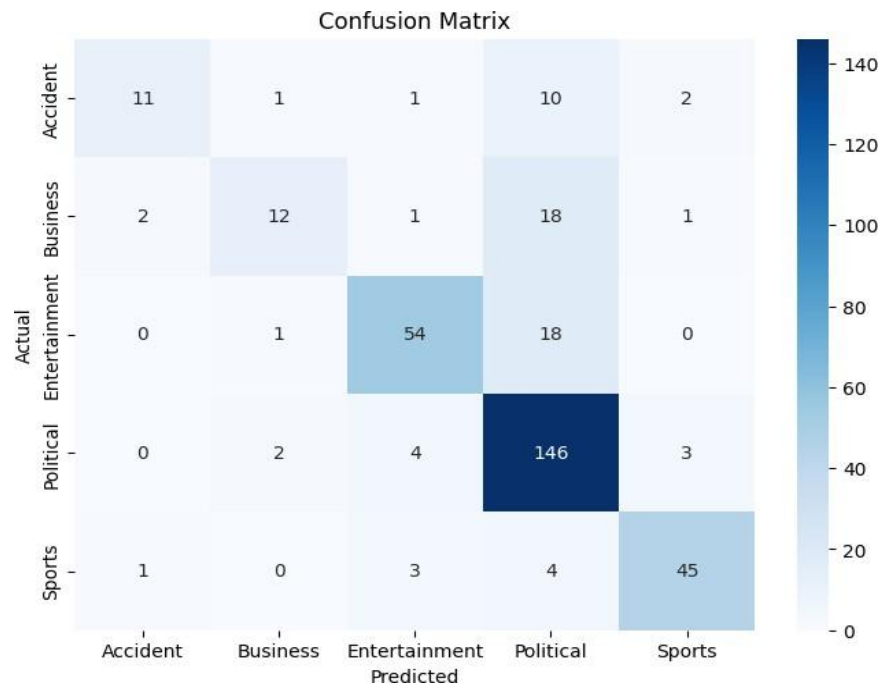


Figure 5.5 Confusion Matrix of HLM-CLS with FastText for 40 Epoch



Figure 5.6 Training vs Validation Loss of HLM-CLS with FastText for 40 Epoch

The above shown Figure 5.5 Confusion Matrix suggests that the model performs well overall. The model has high accuracy in predicting politics for 40 epochs including 146 correct prediction and good performance in entertainment. However, the model struggles with the misclassification between business and politics categories.

The above plot Figure 5.6 shows with FastText, the validation loss plateaus and modestly rises after a few epochs, while the training loss smoothly drops and approaches zero. This suggests that there is a discrepancy between training and validation performance because the model begins overfitting after initial training.

5.3 COMPARATIVE ANALYSIS

Based on the above analysis of HNVec (Hindi News Vectorizer), TF-IDF (Term Frequency-Inverse Document Frequency) and FastText embeddings integrated with HLM-CLS, Hnvec demonstrates the superior performance compared to traditional TF-IDF representations and FastText embeddings since it balance the precision, recall, F1-score while achieving the best test accuracy.

When TF-IDF is integrated with HLM-CLS it causes sparsity issues and also lacks deep semantics understandings whereas fasttext lacks at handling subword information meanwhile HNvec effectively captured the semantics and contextual nuances of hindi news article leading to better feature representation for classification task. As a result, model train on HNVec is found to be an efficient technique for Hindi News Classification. It is a tailored context dependent word embedding utilising the distance based co-occurrence matrix, attain best result in 70:30 train:test split with 88.61% test accuracy and 94.52% train accuracy.

5.4 SUMMARY

Precision, Recall, and F1 Score are essential parameters to judge the system's performance. Precision is the number of correctly translated words to output length; Recall is the number of correctly translated words to reference length; F1 score is the harmonic mean of precision and recall. Integrating HNVec (Hindi News Vectorizer) with CNN-LR-SVM model is the most effective vectorizer that transforms raw text into numerical vectors, since it balances Recall, Precision, and F1 Score while achieving the best test accuracy.

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 INTRODUCTION

As discussed earlier, the entire research is arranged and documented in different chapters. In Chapter 1, a brief introduction to the research work is presented. Chapter 2 discusses and literature available in a similar area of the research. Based on that literature, the objectives of the research work are identified. Chapter 3 is about the theoretical analysis of the machine learning and deep learning models, and how ML integrate with NLP for better classification. All substrate materials used in the research are discussed in Chapter 4. After that, in the same chapter, the methodology for developing hindi news classification system using various models. Chapter 5 discusses the results of the proposed two models for classification system and its performance in terms of accuracy. Outline of proposed work and the future scope, are discussed in sections 6.2 and 6.3 respectively. Finally, the chapter is summarized in section 6.4.

6.2 OUTLINE OF PROPOSED WORK

Text classification is particularly vital due to the abundance of uncategorized data. The surge in content on Hindi news platforms calls for better automated methods to organize and access it. This study aims to instigate Deep learning, Natural Language Processing and Machine Learning approaches to classify Hindi news articles efficiently. The method implemented is to create a scalable and effective hybrid learning framework that combines deep learning and conventional machine learning models with a unique vectorization technique.

The present study introduces HNVec (Hindi News Vectorizer), a domain-specific embedding model tailored for Hindi news articles. HNVec combines efficient distance-based co-occurrence matrix computation with Hindi-specific preprocessing and vocabulary mapping strategies. Unlike generic embedding tools, HNVec is designed to capture the morphological richness and syntactic structure of Hindi, enabling improved classification performance. This model serves as both an embedding generator and a vectorizer, optimized for the challenges of processing low-resource, morphologically rich languages such as Hindi.

Three different vectorization techniques i.e. a tailored context dependent word embedding vectorizer (HNVec), Term Frequency-Inverse Document Frequency (TF-IDF), FastText embeddings, are used to systematically test the hybrid CNN-LR-SVM architecture in order to examine the effects of various textual representations on model performance. in terms of accuracy,

precision, F1 score, and recall, achieving a better accuracy rate.

6.3 FUTURE SCOPE OF WORK

The research work inspires future research directions and potential extensions. The development of Hindi news classification models has vast potential, especially with the increasing demand for regional language content and advancements in machine learning (ML) and natural language processing (NLP). The different directions described that may explore future research works are:

- Develop models capable of classifying news in Hindi-English mixed text (Hinglish) to cater to the growing prevalence of mixed-language usage in digital platforms.
- Move beyond generic categories (e.g., sports, politics) to subcategories like cricket, football, or elections.
- Create specialized models for niche domains like agriculture, regional development, or social issues.
- Focus on tagging articles related to specific types of events, such as natural disasters, economic trends, or festivals.
- Use classification models to deliver personalized news recommendations based on user interests, sentiment preferences, or geographic location.
- Combine classification models with user behavior analysis to predict intent and suggest relevant content.

6.4 SUMMARY

This chapter discusses the outline of the entire research work. In the introduction section of the chapter, a brief overview of all the chapters is presented. After that outline of proposed work and the future scope of this research work is discussed, leading to the future extension of the work carried out in this research.

REFERENCES

- [A]. Jain, V., & Kashyap, K. L. (2023). "Ensemble hybrid model for Hindi COVID-19 text classification with metaheuristic optimization algorithm". *Multimedia Tools and Applications* 82:16839–16859 <https://doi.org/10.1007/s11042-022-13937-2>.
- [B]. Krishnamoorthy, P., Sathiyarayanan, M., & Proença, H. P. (2024). A novel and secured email classification and emotion detection using hybrid deep neural network. *International Journal of Cognitive Computing in Engineering*. <https://doi.org/10.1016/j.ijcce.2024.01.002>.
- [C]. Dodda, R., & Alladi, S. B. (2024). Enhancing document clustering with hybrid recurrent neural networks and autoencoders: A robust approach for effective semantic organization of large textual datasets. *EAI Endorsed Transactions on Intelligent Systems and Machine Learning Applications*, 1(1).
- [D]. Livieris, I.E., Iliadis, L., & Pintelas, P. (2020). On ensemble techniques of weight-constrained neural networks. *Evolutionary Systems*, 1–13. <https://doi.org/10.1007/s12530-019-09331-w>.
- [E]. Mohammed, A., & Kora, R. (2022). An effective ensemble deep learning framework for text classification. *Journal of King Saud University – Computer and Information Sciences*, 34, 8825–8837. <https://doi.org/10.1016/j.jksuci.2021.11.001>
- [F]. Sahoo Kumar Sovan, Saha Saumajit, Ekbal Asif, Bhattacharyya Pushpak and Mathew Jimson (2019) "Event-Argument Linking in Hindi for Information Extraction in Disaster Domain" *CICLing 2019*.
- [G]. Ahmad Zishan, Sahoo Kumar Sovan, Ekbal Asif, Bhattacharyya Pushpak (2018) "A Deep Learning Model for Event Extraction and Classification in Hindi for Disaster Domain" *Proc. of ICON-2018*, Patiala, India. December 2018 c2018 NLP AI, pages 127–136.
- [H]. Shah, K., Patel, H., Sanghvi, D. et al. (2020). A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification. *Augment Hum Res* 5, 12 (2020). <https://doi.org/10.1007/s41133-020-00032-0>.
- [I]. H. Alzoubi, Yehia Ibrahim, Ahmet E. Topcu, and Ahmed Enis Erkaya. (2023). "Machine Learning-Based Text Classification Comparison: Turkish Language Context" *Applied Sciences* 13, no. 16: 9428. <https://doi.org/10.3390/app13169428>.
- [J]. Sitaula, C., Shahi, T.B. (2024). Multi-channel CNN to classify Nepali COVID-19 related tweets using hybrid features. *J Ambient Intell Human Comput* 15, 2047–2056 (2024). <https://doi.org/10.1007/s12652-023-04692-9>

- [K]. Jain, V., Kashyap, K.L. (2023). Ensemble hybrid model for Hindi COVID-19 text classification with metaheuristic optimization algorithm. *Multimed Tools Appl* 82, 16839–16859 (2023). <https://doi.org/10.1007/s11042-022-13937-2>
- [L]. Mundra, S., Mittal, N. (2022). FA-Net: fused attention-based network for Hindi English code-mixed offensive text classification. *Soc. Netw. Anal. Min.* 12, 100 (2022). <https://doi.org/10.1007/s13278-022-00929-1>
- [M] Lei Zhang, Shuai Wang, and Bing Liu (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 4 (2018), e1253.
- [N] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). Deep contextualized word representations. In *Proceedings of NAACL-HLT*. 2227–2237.
- [O] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher (2017). Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*. 6294–6305.
- [P] Alan Akbik, Duncan Blythe, and Roland Vollgraf (2018). Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*. 1638–1649.
- [Q] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [R] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* (2019).
- [S] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2017). Efficient Estimation of Word Representations in Vector Space. *arXiv Preprint*.
- [T] Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (1997). A Neural Probabilistic Language Model. *Neural Information Processing Systems (NIPS)*.
- [U] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2015). Enriching Word Vectors with Subword Information. *arXiv Preprint*. 1
- [V] Srivastava, S., & Pandey, N. (2018). Word Embeddings for Hindi News Classification. *ProQuest Dissertations & Theses Global*.