MULTILINGUAL SENTIMENT TEXT ANALYSIS.

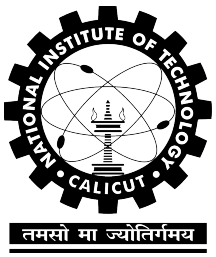
CS4099D Project
End Semester Report

Submitted by

Jammigumpala Sainadh (B190347CS)
Pagala Sanath Reddy (B190182CS)
Shaik Rabnawaz (B191139CS)

Under the Guidance of

Dr.Raju Hazari
Assistant Professor

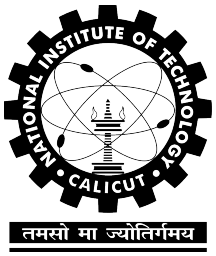


Department of Computer Science and Engineering
National Institute of Technology Calicut
Calicut, Kerala, India - 673 601

May 2023

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
KERALA, INDIA - 673 601

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

Certified that this is a bonafide report of the project work titled

MULTILINGUAL SENTIMENT TEXT ANALYSIS.

done by

Jammigumpala sainadh
Shaik Rabnawaz
Pagala sanath Reddy

of Eighth Semester B. Tech, during the Winter Semester 2022-'23, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering of the National Institute of Technology, Calicut.

(Dr. Raju Hazari)
(Assistant Professor)

04-05-2023
Date

Project Guide

DECLARATION

I hereby declare that the project titled, **MULTILINGUAL SENTIMENT TEXT ANALYSIS**, is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or any other institute of higher learning, except where due acknowledgement and reference has been made in the text.

Place : NIT Calicut
Date : 04-05-2023

Name : Jammigumpala Sainadh
Roll. No. : B190347CS

Name : Shaik Rabnawaz
Roll. No. : B191139CS

Name : Pagala Sanath Reddy
Roll. No. : B190182CS

Abstract

Sentiment analysis is the process of extracting emotions or opinions from a piece of text about a given topic. It helps us understand the attitudes, convictions, and sentiments expressed in the text. It gathers user preferences from web content. It involves predicting or analyzing the hidden information present in the text. This hidden information is beneficial for gaining insights into users' preferences. The goal of sentiment analysis is to determine the writer's or speaker's or consumer's attitudes or opinions towards a given topic or product. The main objective of this project is to implement a model that is capable of analyzing the sentiments of various different texts taken from various social media platforms in different Indian or European languages.

ACKNOWLEDGEMENT

We would like to express our sincere appreciation to Dr. Subhashree M, the Head of Computer Science and Engineering Department at NIT Calicut, for granting us the opportunity to work on this project. Our project coordinator, Dr. Vasudevan AR, deserves our heartfelt gratitude for his dedicated efforts in organizing the project milestones. We are also immensely thankful to our guide and mentor, Dr. Raju Hazari and Ms.Elizabeth M J, for their active guidance, help, cooperation, and encouragement throughout the final year project. Without their support, we would not have progressed in the project. We extend our thanks to our parents and faculty members for motivating and supporting us throughout our work. We also acknowledge the assistance provided by the staff of the CSE Department at NIT Calicut. Additionally, we would like to express our gratitude to our friends and seniors who cooperated with us during the project's course.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

When making daily decisions, we frequently seek out other people's perspectives. We read customer reviews and seek advice from friends when deciding which appliance to purchase. Nowadays, the Internet makes it possible to research the opinions of millions of people on a variety of topics, from the newest technological advancements to political ideologies. Just under one in five internet users (19%), according to the most recent Pew study on the Internet and Civic Engagement, have posted content about political or social issues or used social networking sites for some kind of civic or political engagement.The World Wide Web is quickly developing into a place where people can discuss topics and a source of information for an increasing number of people. Because opinionated text is so common, a new subfield of text analysis has emerged that expands the field's traditional emphasis on information and facts to include applications that are sentimentally aware. The extraction of sentiment from text has received significant attention in the last ten years from both industry and academia. Businesses are becoming more and more aware of the value of online reviews of their goods and services.

Sentiment analysis systems are utilized in nearly every industry and social setting because opinions are at the core of almost all human endeavors and are a significant influence on our actions. Our opinions, worldviews, and

decisions are heavily influenced by how other people perceive and assess the world. For this reason, when we need to make a decision, we often seek out the opinions of others. Not just for people, but also for organizations, this is true and organizations tend to analyze the data for better developing their services and products, When analyzing product reviews or service comments, it is possible to come across comments in a language that is unfamiliar to the analyst. Therefore, it is important to develop a model that can overcome the language barrier and is language-independent, while also being able to perform sentiment analysis.

The majority of the systems used today for sentiment analysis only deal with one language, typically English. On the other hand, users now post comments in various languages due to the global expansion of the Internet. When only one language is used for sentiment analysis, there is a greater chance that crucial information in texts written in other languages will be missed so there is a need for developing sentiment analysis models that are independent of language. The research on multilingual sentiment analysis that involves the development of a single model capable of identifying emotions across various languages is not well-established or conclusive. It is necessary to create a model that can identify the sentiment in not just English, but also in various Indian and foreign languages.

Chapter 2

Problem Statement

The main objective of this project is to implement a model that is capable of predicting the sentiments of various different texts taken from various social media platforms in different languages to be labeled as Positive or Negative. As the existing systems mainly focus on English language sentiment analysis, this model can be extendable to other languages.

2.1 Objectives

This project aims to achieve the following objectives:

- To design and implement a model that is capable of analyzing the sentiment of the textual data.
- The designed model must be able to predict Sentiment analysis over multiple languages.

Chapter 3

Literature Survey

In 2021, M.J. Elizabeth Et al.[5]introduced a new text classification approach, which combines cellular automata (CA) and machine learning techniques. The approach involves using a CA model to preprocess text data and extract important features, which are then fed into a machine learning model for classification. The approach was specifically applied to the task of toxic text classification, which aims to identify toxic comments or posts on social media platforms. The authors evaluated their approach on various benchmark datasets and compared it with existing state-of-the-art methods. The experimental results demonstrated that the proposed approach achieved high accuracy in identifying toxic text, surpassing existing methods. The authors suggested that their approach could have practical applications in fields such as cybersecurity and content moderation on social media platforms, and that it could also be applied to other text classification tasks.

In 2019, Saad Et al.[11] proposed a sentiment analysis model that employs machine learning algorithms based on ordinal regression to perform comprehensive sentiment analysis of tweets. The model consists of four main modules that work together to achieve the task.

In the first module, the model collects labeled tweets that will be used

for sentiment analysis. In the second module, the collected dataset is pre-processed to improve the quality of the data and make it suitable for further analysis. In the third module, relevant features are extracted from the pre-processed dataset, which are then used to create a classification model. The tweets are then balanced and scored using a particular method.

In the final module, the model uses various machine learning algorithms such as Support Vector Regression (SVR), Random Forest (RF), Multinomial Logistic Regression (Soft Max), and Decision Trees (DTs) to classify tweets into different categories based on their sentiment. The categories include high positive, moderate positive, neutral, moderate negative, and high negative. This way, the sentiment of tweets can be analyzed and categorized accordingly. The proposed model shows promising results in analyzing the sentiment of texts and classifying them based on their emotional tone.

In 2019,Aditya Et al. [8] presented his work on sentiment analysis in Hindi, which involved three different approaches. In the first approach, they trained and tested the classifier on Hindi language documents, which is referred to as In-language sentiment analysis. The second approach was Machine Translation (MT) - based sentiment analysis, where the classifier was trained on English documents, and the Hindi documents were translated to English using a translation module before being classified. In the third approach, they used a majority-based sentiment classifier based on H-SWN, a lexical resource called Hindi-SentiWordNet. They conducted experiments with different variants, such as with or without stop word removal and stemming, and with the scores of all senses or only the most common sense of a word considered for polarity determination.

The results of the experiments showed that the In-language sentiment analysis approach outperformed the other two approaches, with an accuracy of 78.14. The accuracy for MT-based sentiment analysis was 65.96, while Resource-based sentiment analysis had an accuracy of 60.31. These results

indicate that using an annotated corpus in the same language as the analysis can lead to the best results in sentiment analysis, and that Machine Translation may not always be reliable in preserving the sentiment of a text during translation. The use of lexical resources like H-SWN(lexical resource called Hindi-SentiWordNet) can also be helpful, but may not always provide accurate results.

In 2019, Ray Et al. [10] proposed a novel deep learning-based algorithm for feature extraction and sentiment analysis of user opinions from text. The authors utilized a seven-layer convolutional neural network (CNN) to tag the features in opinionated sentences. To further improve the accuracy of the aspect extraction process, a combination of techniques such as POS tagging, dependency parsing using CoreNLP, and hierarchical clustering were employed to identify the aspects.

The authors compared their proposed method with existing approaches such as CoreNLP+Rule-based and CNN-based aspect extraction. The results showed that while the CNN-based approach performed better than CoreNLP+Rule-based in most cases, it still failed to identify some valid aspect terms. To address this limitation, a rule-based approach was combined with the CNN-based approach to further improve the performance of the aspect extraction method.

Finally, the suggested method achieved the best accuracy in aspect extraction and sentiment analysis, outperforming the other two methods. The accuracy of CNN+Rule-based, CNN-only, and CoreNLP+Rule-based approaches were reported to be 80%, 75% and 87%, respectively. The study highlights the potential of deep learning-based approaches in improving the accuracy of feature extraction and sentiment analysis tasks in natural language processing.

In 2022, Roobae Alroobaea's [1]In this study, a technique that employed Re-

current Neural Networks (RNN) was proposed for predicting the sentiment of Amazon reviews. The study also involved comparing various types of neural networks with the proposed RNN model. The technique comprised of preprocessing, word embedding, and classification steps. The preprocessing step was done to ensure the quality of the dataset, while word embedding was used to convert words into numerical vectors in a reduced dimensional space. In the classification step, the RNN model was used to predict the sentiment of the authors based on the word embedding vectors, which represented words and their contexts. The results of the RNN model were compared with those of three other well-known deep learning models, and they were found to be better than the results of the Convolutional Neural Network (CNN) model. However, the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models did not perform as well. The accuracy of the LSTM, GRU, CNN, and RNN models were found to be 76, 80, 83, and 85, respectively.

In 2018, Mukhtar et al. [9] conducted sentiment analysis on Urdu blogs using two different approaches: Lexicon-based models and Supervised Machine learning algorithms. In the Lexicon-based model, an Urdu sentiment analyzer and an Urdu Sentiment Lexicon were used to classify the sentiment of the blogs. The supervised machine learning algorithms used were Decision Trees (DT), K-Nearest Neighbors (KNN), and Support Vector Machines (SVM). The data from both approaches were combined to achieve the best sentiment analysis results.

After conducting several tests, the author found that the Lexicon-based model performed better than the supervised machine learning algorithms. This suggests that the sentiment of Urdu blogs can be accurately determined using a lexicon-based approach rather than using machine learning algorithms.

In 2019, Vashishtha et al. [14] have developed a new approach for sentiment

analysis of social media posts by combining Word Sense Disambiguation, natural language processing (NLP) models, and an unsupervised fuzzy rule-based model. This approach involved the use of a new set of fuzzy rules that incorporated multiple datasets and lexicons. The aim was to categorize the comments into three sentiment classes: negative, neutral, and positive. The experiments were conducted on nine freely available Twitter datasets and compared against three sentiment lexicons and four existing models. The results showed that the proposed approach achieved the best results compared to the other methods. Overall, this study provides a promising new approach for sentiment analysis of social media posts.

In 2020, Xu et al. [15]proposed a Naive Bayes (NB) method for sentiment classification of product reviews in large-scale E-Commerce platforms across multiple domains. To improve the model's performance, they extended the parameter evaluation method in NB to a continuous learning fashion. They also introduced several approaches for fine-tuning the learned distribution based on three different assumptions. The proposed method was evaluated on Amazon product and movie review datasets, and achieved high accuracy in sentiment classification.

In 2019, Bardhan et al. [2] utilized a quasi-qualitative model to investigate the impact of gender mainstreaming in Slum Rehabilitation Housing (SRH) management. The researchers conducted semi-structured interviews and focused group discussions to understand the concerns of stakeholders. To analyze the emotions of stakeholders, they employed sentiment analysis using a machine learning algorithm based on natural language processing (NLP).In this study, sentiment analysis was conducted using a Unigram-based approach, which involved assigning a polarity score to each individual word, and then calculating the overall score for the analyzed text by summing up the scores of all the words. To obtain the overall sentiment, a simple term counting method

was utilized. Specifically, the number of positive words was subtracted from the number of negative words to derive a general polarity score.

In 2021, Santosh Kumar Dubey et al. [6] work proposes an integrated CNN-based approach for sentiment analysis of Hindi tweets, a scarce-resource language. The proposed method combines convolutional neural network (CNN) and word embedding techniques for feature extraction and classification. The study also employs data augmentation and transfer learning to improve the accuracy of the classification model. The performance of the proposed model was evaluated using standard metrics such as accuracy, F1 score, and confusion matrix. The results demonstrate that the proposed model outperforms several baseline models, achieving an accuracy of 85%. The study concludes that the proposed approach could be useful for sentiment analysis of other scarce-resource languages.

In 2022,Elizabeth M J et al . [4]The objective of this research paper is to identify toxic comments in social media using a unique approach based on cellular automata and LSTM (Long Short-Term Memory) model. The proposed model does not depend on language, and it is able to achieve high accuracy in identifying toxic comments. The model does not require the use of any pre-trained word embeddings or language models. The use of cellular automata in the proposed approach allows the model to incorporate the neighboring characters of a word and consider their impact on the sentiment analysis. The model was tested on a large dataset of comments from social media platforms and was able to achieve an Accuracy of 97.43%, indicating its high accuracy in identifying toxic comments. Overall, the proposed model can be a valuable tool in detecting toxic comments and promoting a healthy and positive online community.

In 2019 ,Thongtan et al . [13]experiments on the cosine similarity in the year 2019 Using feature combination and Nave Bayes, the IMDB dataset demonstrates that accuracy is increased when cosine similarity is used instead of the dot product.

The 2021,Samia et al. [12] paper "Aspect-based Sentiment Analysis for Bengali Text using Bidirectional Encoder Representations from Transformers (BERT)" presents a study on ABSA for Bengali language using BERT. The authors fine-tune the BERT model for ABSA on Bengali text and evaluate its performance on a dataset of Bengali product reviews. They show that their BERT-based approach outperforms several baseline models in terms of accuracy and F1 score. The paper concludes by highlighting the effectiveness of BERT-based models for ABSA in Bengali text and suggests future work in expanding the dataset and exploring different transformer-based models for the task. In 2022, Hazari et. al.[7] presented An analysis of the coronavirus envelope protein was performed using a cellular automata (CA) model.. The research focused on examining the physicochemical characteristics of a particular protein and its interactions with potential medicinal agents. The authors utilized a cellular automata (CA) model to simulate the folding and unfolding processes of the protein and to investigate its behavior in different conditions. The results indicated that the protein exhibited significant structural stability and retained its structure even in the presence of strong denaturants. The study also identified several possible binding sites on the protein that could be targeted by small molecule drugs or antibodies. According to the authors, the CA model employed in this research has the potential to be applied to other proteins and could contribute to the development of new treatments and drugs for infectious diseases.

In 2022, Partha Chakraborty, et.al [3] published a paper titled "Comparative Analysis of Classical and Deep Learning Approaches in Sentiment Detection

of Bengali Text on Facebook." The study proposes a sentiment analysis approach for detecting the polarity of Bengali Facebook posts and comments using seven machine learning algorithms, including five classical and two deep learning approaches. After preprocessing the raw data and applying the TF-IDF technique for feature extraction, the study compared the performance of the classifiers used in sentiment detection. The results indicate that deep learning approaches outperformed classical approaches, with an accuracy of 96.95

Chapter 4

Proposed Work

The main goal of the models presented in this report is to develop an effective and efficient method for identifying the relevant sentiments expressed in textual data, regardless of the language in which the text is written. In this chapter, we provide a comprehensive overview of the design of our approach, which encompasses our methodology for annotating the data, the architecture of our system, and our approach to feature extraction.

The primary objective of this models are to build a language-independent system that can accurately classify sentiments in a wide range of textual data. To achieve this, we focus on developing an innovative and robust approach that incorporates various natural language processing techniques, machine learning algorithms, and feature engineering strategies, then describe the architecture of our system, which consists of several key components, including a data preprocessing module, a feature extraction module, and a machine learning module.

4.1 Proposed models

Here two models are proposed based on the concepts of cellular automata and machine learining .

The two models are as follows

- Rule Based Model.
- Mutations Based Model.

4.2 Basics of Cellular Automata

Cellular Automata (CA) is a concept in computer science that involves a grid of cells, each of which can exist in a particular state. The states of the cells are updated over time according to a set of rules that depend on the states of neighboring cells. The rules are applied simultaneously to all cells in the grid. The concept is often used to model complex systems that exhibit emergent behavior, such as natural phenomena or social dynamics.

4.2.1 Elementary Cellular Automata

A Cellular Automata(CA) consists of a grid-like structure made up of individual cells, each of which stores a value representing its current state at a given time 't'. The next state of a cell is determined based on its current state and the states of its neighboring cells at the same time 't'.

Elementary Cellular Automata (ECA) is the most basic form of CA and was proposed by Wolfram. In ECA, each cell stores a binary state, and its next state at time 't+1' is determined by a next-state function, denoted by f. This function takes into account the cell's current state, as well as the states of its two closest neighbors. The present states of the left, self, and right neighbors of the i-th cell at time 't' are used as input to determine the next state of the i-th cell at time 't+1'.

$$S_i^{t+1} = f(S_{i-1}^t, S_i^t, S_{i+1}^t) \tag{4.1}$$

where f is the next state function, and $S_{i-1}^t, S_i^t and S_{i+1}^t$ are the present states

of the left, self, and right neighbor of the i^{th} cell at time t. In a traditional cellular automaton, all cells follow the same next-state function, making it a uniform CA. However, in a non-uniform or hybrid CA, each cell is allowed to follow a different next-state function or rule. In this study, we utilized a non-uniform elementary cellular automaton (ECA) under periodic boundary conditions, where the first and last cells are considered neighbors.In the case of ECA, the boundary condition is periodic boundary condition, where the first cell is considered a neighbor of the last cell, and vice versa. Overall, cellular automata provide a framework for understanding the behavior of complex systems that change over time based on local interactions between their constituent parts. ECA is a simple yet powerful example of this concept, which can be applied to a variety of fields, including physics, biology, and computer science.

Present State :	111 (7)	110 (6)	101 (5)	100 (4)	011 (3)	010 (2)	001 (1)	000 (0)	*Rule*
(i) Next State :	0	1	0	1	1	0	1	0	90
(ii) Next State :	1	0	0	1	0	1	1	0	150

Figure 4.1: Rule determination

4.3 Rule Based Approach

4.3.1 Overview

A rule-based approach is a method of analyzing data that does not involve training or using machine learning models. This approach relies on explicitly defined rules and logic to make decisions based on predefined conditions. In contrast to machine learning, which requires large amounts of data to train

models, the rule-based approach relies on the expertise of domain experts to define the rules that govern decision making. By using this approach, it is possible to analyze data and make decisions in a practical and straightforward manner, without the need for sophisticated algorithms or large amounts of data. However, it is important to note that the rule-based approach may have limitations in terms of its ability to generalize to new or complex data, and it may require continuous refinement as new information becomes available.

4.3.2 Cellular Automata for rule based sentiment analysis

Cellular automata can be used as a rule-based approach in sentiment analysis by transforming comments into a set of rules based on their ASCII values. These rules can then be used to generate a cellular automaton grid where each cell's state is updated based on its neighboring cells' states and the predefined rules. By allowing the cellular automaton to evolve over time, it is possible to calculate the cycle length (CL) of each cell, and generate a CL graph.

Analyzing these graphs can provide valuable information about the sentiment of the comments. For example, the graphs may reveal patterns that indicate positive or negative sentiment, or they may reveal correlations between certain words or phrases and particular sentiment.

Using cellular automata as a rule-based approach in sentiment analysis has the advantage of being able to capture complex patterns and relationships that may be difficult to identify using other methods.

4.3.3 Cellular Automata for Multilingual sentiment Analysis

One advantage of using cellular automata for multilingual sentiment analysis is that it can be applied to a wide range of languages without the need for

complex language-specific models. However, it is important to consider the differences in character encoding and language syntax when designing the rules and parameters for the cellular automaton. It is also important to evaluate the performance of the approach on a variety of multilingual data sets to ensure its effectiveness.

4.3.4 Design

The design is mainly divided in four modules

- Module-I : Text to rule vector conversion
- Module-II : CL Value generation
- Module-III : Parameter extraction
- Module-IV : Prediction

Module-I : Text to rule vector conversion

The rule vector for the cellular automaton model is generated by taking the ASCII value of each character present in the input text and inserting it into the vector. This means that the rules for the model are derived directly from the characters in the text, with each character being assigned a specific numeric value based on its corresponding ASCII code.

- Input:HE IS A GOOD BOY
- Rule vector:[104, 101, 32, 105, 115, 32, 97, 32, 103, 111, 111, 100, 32, 98, 111, 121]

Character	Unicode	Rule	Character	Unicode	Rule	Character	Unicode	Rule
अ	2309	9, 14	ण	2339	9, 44	◌	2373	9, 78
आ	2310	9, 15	त	2340	9, 45	◌	2375	9, 80
इ	2311	9, 16	थ	2341	9, 46	◌	2376	9, 81
ई	2312	9, 17	द	2342	9, 47	◌	2377	9, 82
उ	2313	9, 18	ध	2343	9, 48	◌	2379	9, 84
ऊ	2314	9, 19	न	2344	9, 49	◌	2380	9, 85
ऋ	2315	9, 20	प	2346	9, 51	◌	2381	9, 86
ऌ	2316	9, 21	फ	2347	9, 52	◌	2384	9, 89
ऍ	2317	9, 22	ब	2348	9, 53	।	2404	9, 109
ए	2319	9, 24	भ	2349	9, 54	॥	2405	9, 110
ऐ	2320	9, 25	म	2350	9, 55	०	2406	9, 111
ऑ	2321	9, 26	य	2351	9, 56	१	2407	9, 112
ओ	2323	9, 28	र	2352	9, 57	२	2408	9, 113
औ	2324	9, 29	ल	2354	9, 59	३	2409	9, 114
क	2325	9, 30	ळ	2355	9, 60	४	2410	9, 115
ख	2326	9, 31	व	2357	9, 62	५	2411	9, 116
ग	2327	9, 32	श	2358	9, 63	६	2412	9, 117
घ	2328	9, 33	ष	2359	9, 64	७	2413	9, 118
ङ	2329	9, 34	स	2360	9, 65	८	2414	9, 119
च	2330	9, 35	ह	2361	9, 66	९	2415	9, 120
छ	2331	9, 36	◌	2364	9, 69	॰	2416	9, 121
ज	2332	9, 37	ऽ	2365	9, 70			
झ	2333	9, 38	◌	2366	9, 71			
ञ	2334	9, 39	◌	2367	9, 72			
ट	2335	9, 40	◌	2368	9, 73			
ठ	2336	9, 41	◌	2369	9, 74			
ड	2337	9, 42	◌	2370	9, 75			
ढ	2338	9, 43	◌	2371	9, 76			

Figure 4.2: Hindi-Rules

Module-II : CL Value generation

This module generates CL value sequences using rule vectors, which are used to initialize a cellular automaton (CA) with alternating 1's and 0's. The length of this sequence is equal to the length of the rule vector. Using the rule number in the rule vector, the next state of the current cell is determined based on the states of its neighboring cells in the CA. This process is repeated for 1000 iterations, which is chosen based on the length of the text being analyzed.

Character	Unicode	Rules	Character	Unicode	Rules	Character	Unicode	Rules	Character	Unicode	Rules	Character	Unicode	Rules
	32	32	§	167	167	ড	2465	9, 170	ে	2503	9, 208	‘	8216	32, 56
!	33	33	©	169	169	ঢ	2466	9, 171	ৈ	2504	9, 209	’	8217	32, 57
"	34	34	ঁ	2433	9, 138	ণ	2467	9, 172	ো	2507	9, 212	“	8220	32, 60
#	35	35	ং	2434	9, 139	ত	2468	9, 173	ৌ	2508	9, 213	”	8221	32, 61
$	36	36	ঃ	2435	9, 140	থ	2469	9, 174	্	2509	9, 214	†	8224	32, 64
%	37	37	অ	2437	9, 142	দ	2470	9, 175	ৎ	2510	9, 215	‡	8225	32, 65
&	38	38	আ	2438	9, 143	ধ	2471	9, 176	ৗ	2519	9, 224	…	8230	32, 70
'	39	39	ই	2439	9, 144	ন	2472	9, 177	ড়	2524	9, 229	‰	8240	32, 80
(	40	40	ঈ	2440	9, 145	প	2474	9, 179	ঢ়	2525	9, 230	′	8242	32, 82
)	41	41	উ	2441	9, 146	ফ	2475	9, 180	য়	2527	9, 232	″	8243	32, 83
*	42	42	ঊ	2442	9, 147	ব	2476	9, 181	ৠ	2528	9, 233	€	8364	32, 204
+	43	43	ঋ	2443	9, 148	ভ	2477	9, 182	ৡ	2529	9, 234			
,	44	44	ঌ	2444	9, 149	ম	2478	9, 183	ৢ	2530	9, 235			
-	45	45	এ	2447	9, 152	য	2479	9, 184	ৣ	2531	9, 236			
.	46	46	ঐ	2448	9, 153	র	2480	9, 185	০	2534	9, 239			
/	47	47	ও	2451	9, 156	ল	2482	9, 187	১	2535	9, 240			
:	58	58	ঔ	2452	9, 157	শ	2486	9, 191	২	2536	9, 241			
;	59	59	ক	2453	9, 158	ষ	2487	9, 192	৩	2537	9, 242			
<	60	60	খ	2454	9, 159	স	2488	9, 193	৪	2538	9, 243			
=	61	61	গ	2455	9, 160	হ	2489	9, 194	৫	2539	9, 244			
>	62	62	ঘ	2456	9, 161	়	2492	9, 197	৬	2540	9, 245			
?	63	63	ঙ	2457	9, 162	ঽ	2493	9, 198	৭	2541	9, 246			
@	64	64	চ	2458	9, 163	া	2494	9, 199	৮	2542	9, 247			
[	91	91	ছ	2459	9, 164	ি	2495	9, 200	৯	2543	9, 248			
\	92	92	জ	2460	9, 165	ী	2496	9, 201	৺	2554	10, 4			
]	93	93	ঝ	2461	9, 166	ু	2497	9, 202	-	8208	32, 48			
^	94	94	ঞ	2462	9, 167	ূ	2498	9, 203	-	8209	32, 49			
_	95	95	ট	2463	9, 168	ৃ	2499	9, 204	–	8211	32, 51			
|	124	124	ঠ	2464	9, 169	ৄ	2500	9, 205	—	8212	32, 52			

Figure 4.3: Bengali-Rules

After a certain number of iterations, a pattern starts repeating in the CA. The length of this pattern is referred to as the Cycle Length (CL) value. A particular CA cell is assigned a CL value if the same pattern is repeated at least 32 times. If no repeated patterns are found in a cell, its CL value is assigned as ‘-5’.

The choice of 32 as the minimum number of repeated patterns required to assign a CL value is arbitrary, as any other value could be used. The CL value represents the length of the repeated pattern in the CA cell. Similarly, the use of ‘-5’ to represent the CL value of a cell with no repeated patterns is also arbitrary, as any negative number could be used instead.

- input:i have a new hand bag its fabulous
- output:5,1,1,1,1,1,1,2,1,1,1,1,1,1,1,1,1,1,12,12, 6,6,6,6,3,3,3,3,3,3,3,15,15,5

Character	Unicode	Rule	Character	Unicode	Rule	Character	Unicode	Rule
A	65	65	]	93	93	z	122	122
B	66	66	^	94	94	|	124	124
C	67	67	_	95	95	¡	161	161
D	68	68	a	97	97	§	167	167
E	69	69	b	98	98	©	169	169
F	70	70	c	99	99	«	171	171
G	71	71	d	100	100	»	187	187
H	72	72	e	101	101	¿	191	191
I	73	73	f	102	102	Á	193	193
J	74	74	g	103	103	É	201	201
K	75	75	h	104	104	Í	205	205
L	76	76	i	105	105	Ñ	209	209
M	77	77	j	106	106	Ó	211	211
N	78	78	k	107	107	Ú	218	218
O	79	79	l	108	108	Ü	220	220
P	80	80	m	109	109	á	225	225
Q	81	81	n	110	110	é	233	233
R	82	82	o	111	111	í	237	237
S	83	83	p	112	112	ñ	241	241
T	84	84	q	113	113	ó	243	243
U	85	85	r	114	114	ú	250	250
V	86	86	s	115	115	ü	252	252
W	87	87	t	116	116			
X	88	88	u	117	117			
Y	89	89	v	118	118			
Z	90	90	w	119	119			
[	91	91	x	120	120			
\	92	92	y	121	121			

Figure 4.4: English-Rules

- input:কোন কুত্তার বাচ্চায় ভিডিওটা ছাড়ছে?
- Output:1,1,1,1,4,2,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,
  1,1,1,2,1,1,1,2,1,1,1,1,1,1,1,1,3,3,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
- input:फिर से दोनों पक्षो में बड़ा भयंकर युद्ध हुआ।
- Output:1,1,1,1,4,2,4,12,12,12,12,1,1,1,1,1,4,2,8,8,8,8,8,4,
  2,1,1,1,1,1,1,1,1,1,1,1,1,1,3,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,

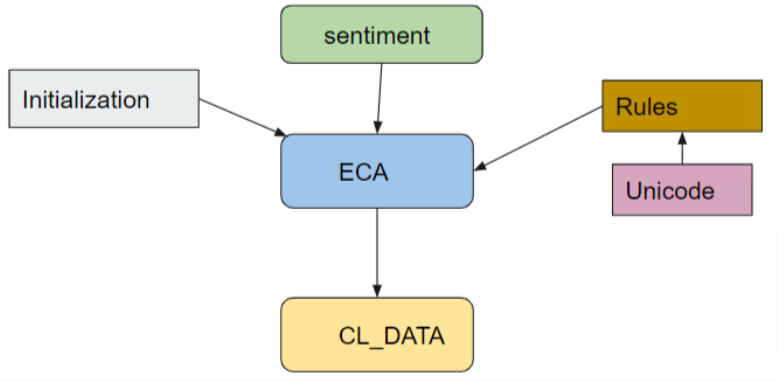


Figure 4.5: CL Generation

3,3,1,1,1,3,3,3,3,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1

Module-III : Parameter extraction

Feature extraction and selection refers to the process of choosing the most important and relevant features from a large set of features to enhance the performance of a machine learning model. The process involves reducing the original set of features into a smaller, more informative subset that can better represent the patterns in the data. The process may use statistical methods, domain knowledge, and machine learning algorithms to identify the most informative and relevant features that can improve the performance of the model while reducing the complexity of the feature space.

- **Maximum cycle length (MCL):**It refers to the highest or largest value among the cycle length sequences.

.....	1	0	0	0	0	0	1	.....
.....	0	0	1	0	1	1	1	.....
.....	0	0	0	1	0	1	1	.....
.....	0	1	1	0	0	1	0	.....
.....	1	0	1	1	0	0	0	.....
.....	0	0	1	0	1	1	1	.....
.....	0	1	0	1	0	1	1	.....
.....	1	0	0	0	0	0	1	.....
.....	0	1	1	1	1	1	0	.....
.....	0	1	1	0	1	1	0	.....
.....	1	0	0	1	0	0	1	.....
.....	0	1	1	0	1	1	1	.....
.....	0	0	0	1	0	0	1	.....
.....	1	0	0	0	0	0	0	.....
.....	0	1	0	1	0	0	0	.....
.....	0	0	0	0	0	0	1	.....
.....	1	1	0	1	1	0	1	.....
.....	0	1	0	0	1	0	1	.....

Figure 4.6: CA Evolution

- **MCL count :**It is determined by identifying the number of peaks in the CL graph that correspond to the maximum cycle length (MCL).
- **NMCL (Next to Maximum Cycle Length):** values are obtained by creating a list of all cycle length values that are less than the MCL (Maximum Cycle Length) value, until a certain threshold value is reached.
- **Right Step:**The right step in the CL graph refers to the horizontal segment of the graph that connects the maximum value of a cycle(MCL) to the subsequent minimum value of the next cycle.
- **Left Step:**It is just like the right step but presented to the left side to the maximum value of a cycle (MCL).

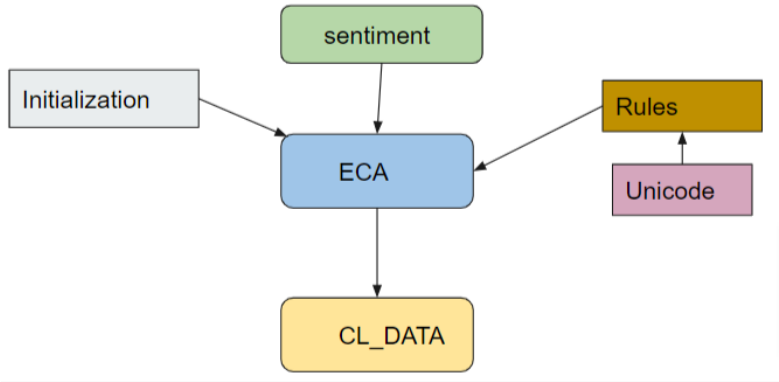


Figure 4.7: CL Generation

- **Fork:** a fork refers to a point where the cycle length sequence splits i.e V shaped line in the CL graph.If fork is present left of MCL then it is **Left fork** else if it is present to the right to the MCL then it is called **Right fork.** The presence or absence of forks and steps can serve as a flag to identify specific types of patterns in the cycle length graph.

- **Filtering into Groups:**The value of the group parameter is determined by considering the maximum cycle length (MCL), the highest next to maximum cycle length (NMCL) values, and the MCL count. The statements are then categorized into three groups, based on the following criteria:

Filter-G0	**Filter-G1**	**Filter-G2**
$MCL - max(NMCL) \leq 20\% of MCL$	$Count(MCL) = 1$	$Count(MCL) > 1$

Table 4.1: Grouping

- **Signal Location Vector:**The signal location vector is a list that includes all MCL and NMCL locations. If there are multiple MCLs, a new signal location vector is created for each one. The new signal lo-

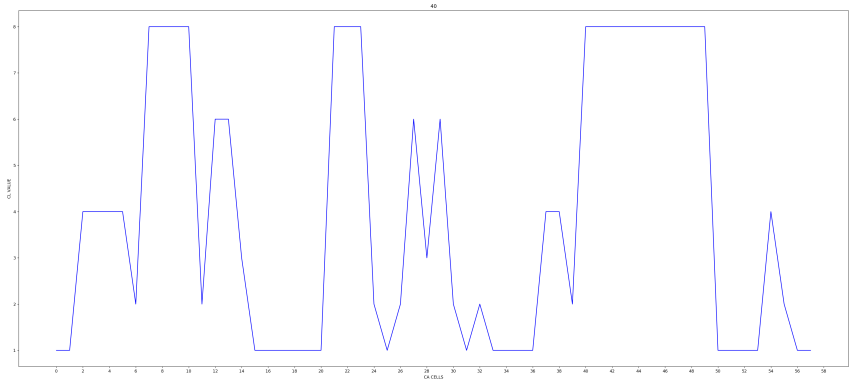


Figure 4.8: CL Graph

cation vector includes all NMCL locations and MCL locations, except for the one that is currently being observed.

- **Signal Distance Vector:**The process involves computing the distance between the current location of MCL and each of the locations listed in the corresponding Signal Location Vector for that MCL.The SDVs are divided into two parts: the Left Signal Distance Vector (LSDV) and the Right Signal Distance Vector (RSDV). In the LSDV, all values to the left of an MCL location in its corresponding SDV are taken, while in the RSDV, all values to the right of an MCL location in its corresponding SDV are taken.

- **Cosine similarity Index:**Cosine similarity index is a measure of similarity between two non-zero vectors of an inner product space. It is the cosine of the angle between the two vectors. CSI is equal to the dot product of the test and train divided with ther lengths.A=train,B=test CSI = (<A, B> /||A||.||B||)

- **Euclidean Distance** The Euclidean distance is a mathematical method to compute the shortest distance between two points in a space that has multiple dimensions. In the context of the present scenario, the Euclidean distance is utilized as a tool to quantify the difference between the CL values of the testing and training data.

- **R-Value** The R value is obtained by dividing the number of matched signal pairs by the total count of mismatches.

  Match Signal Pair: To generate a match signal pair, the distances D-train' in SDV of the training data and D-test' in SDV of the testing data are considered. If the pair (D-train, D-test) satisfies the predetermined conditions, it is considered a match signal pair. This process is repeated for each distance in SDV of the training data and for each distance in SDV of the testing data. The MSP for a given testing and training data is obtained by summing the MSP for SDV.condition as ratio of D-train to D-test.If ratio is more than 0.85 return 1 else 0.

  Mismatch Count: The Mis Match Count (MMC) is incremented by one if none of the training data's SDV forms a matching pair with any of the SDV of testing data

Module-IV:Prediction

The process of using a trained model to make informed guesses or estimations about new, unseen data is known as prediction in machine learning. Another way to describe this is the process of generating an output from a given input using a trained model,Further details are in Results section .

Prediction parameters are as follows:

- Cosine Similarity Index

- Euclidean Distance
- R Value

Algorithm 1 Rule Based algorithm

1: **Input** : $Text$
2: **Output** : *sentiment of Data*
3: *Pre-process each sentence in the dataset for generating rule vectors.*
4: *Convert pre-processed data into CA rule vectors using rule values.*
5: *Find out the CL sequences using CA rule vectors.*
6: *Find the parameters from CL data.*
7: *Extract the selection parameters.*
8: *Apply the extracted selection parameters and filter out the training data for each testing data.*
9: *Compute the predicting parameters, such as CSI, distance metric, and R-value, for every testing data by comparing it with each training data from the chosen cluster.*
10: *Predict their sentiment using the prediction parameters.*
11: *The performance analysis is accuracy using the following formula:*

$$Accuracy = \frac{TotalPositive + TotalNegative}{\text{total \# of sentences in data}} \tag{4.2}$$

4.4 Mutations Based Approach

The Mutation-Based approach involves feeding the original data to an Elementary Cellular Automaton (ECA). The CA generates Cycle Length (CL) data which is saved for future use. The Next Maximum Cycle Length (NMCL) values are calculated from the CL data, and the corresponding NMCL locations are identified and saved as an array of arrays, where each

inner array represents a continuous location of NMCL values. Each NMCL value generates a new mutation.

To mutate the original data, CL value sequences are generated using rule vectors. The CA is initialized with alternate 1's and 0's, and the length of the sequence is the same as the length of the rule vector. Generally, the CL sequence is initialized with alternating 0's and 1's. To mutate the sequence, we take the original NMCL positions data saved in the previous module. We explicitly identify the NMCL positions of the original CL data sequence and initialize them with alternating 0's and 1's for the entirety, except we replace them with all ones at the NMCL location for the 0 to 1 mutated sequence. If the original CL data have multiple NMCL positions, then each NMCL generates a mutated sequence. For example, if a CL data sequence had three NMCL positions, it gives rise to three mutated sequences. We repeat the same process for the 1 to 0 mutated sequence, except instead of replacing all ones at the NMCL position, we replace them with all zeros. We initialize the mutated sequences of the original data into their respective two divisions of the 1 to 0 mutated and 0 to 1 mutated, respectively. We run the process of generating the CL data for every mutated sequence, just as we generated the CL data for the original data generation.

Both divisions of the mutated sequences (0 to 1 and 1 to 0) generate their respective CL data. We calculate certain related conditions that can be used as parameters from the generated mutated CL data. We calculate the CL sum of every mutated CL data and also calculate the CL difference of every mutation with its respective original CL sum without any mutations. By using the cl differences we calculated the parameters

4.4.1 Design

The highlevel desigin of the Mutations based approach shown below in figures 4.9 and 4.10

- Module-I : Module-I refers to the initial phase of the process, which

involves generating the cycle length (CL) data and their corresponding NMCL positions. During this phase, the sum of the CL values is calculated and saved as an additional parameter.

- Module-II:based on the outputs of the module -I the initial sequence is mutated and their corresponding cl data is generated and the vector distance is calculated .

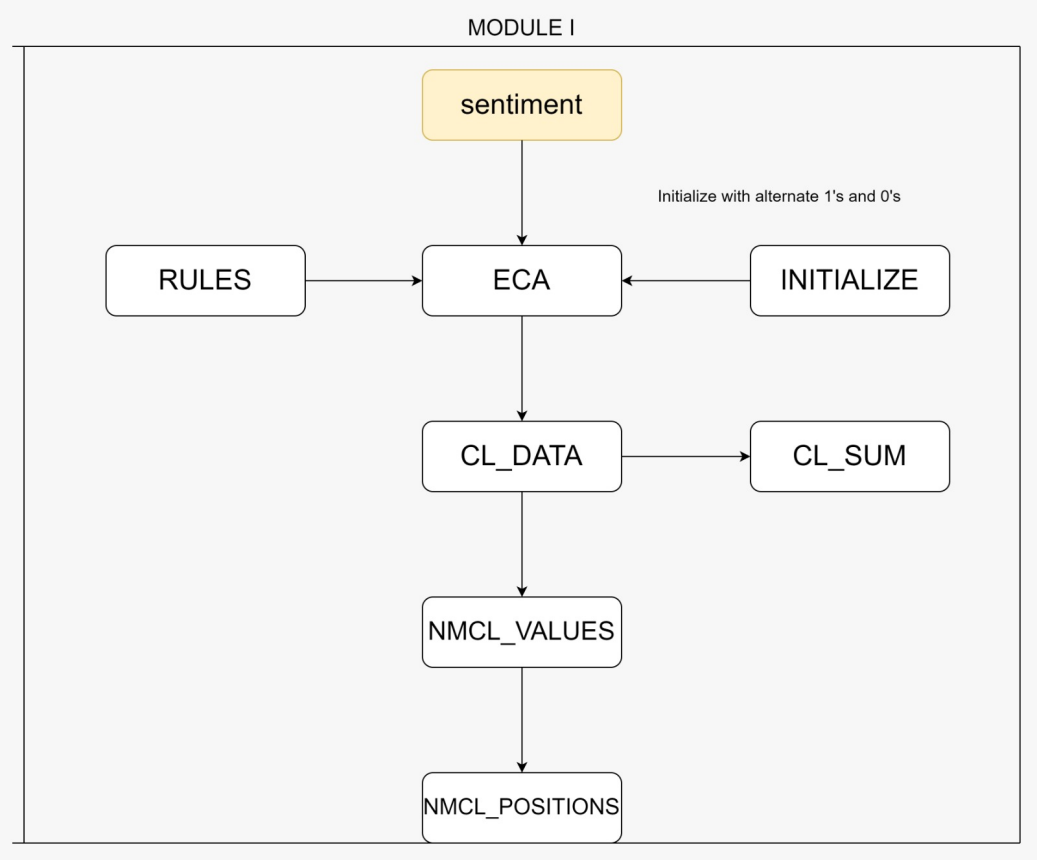


Figure 4.9: Module-I

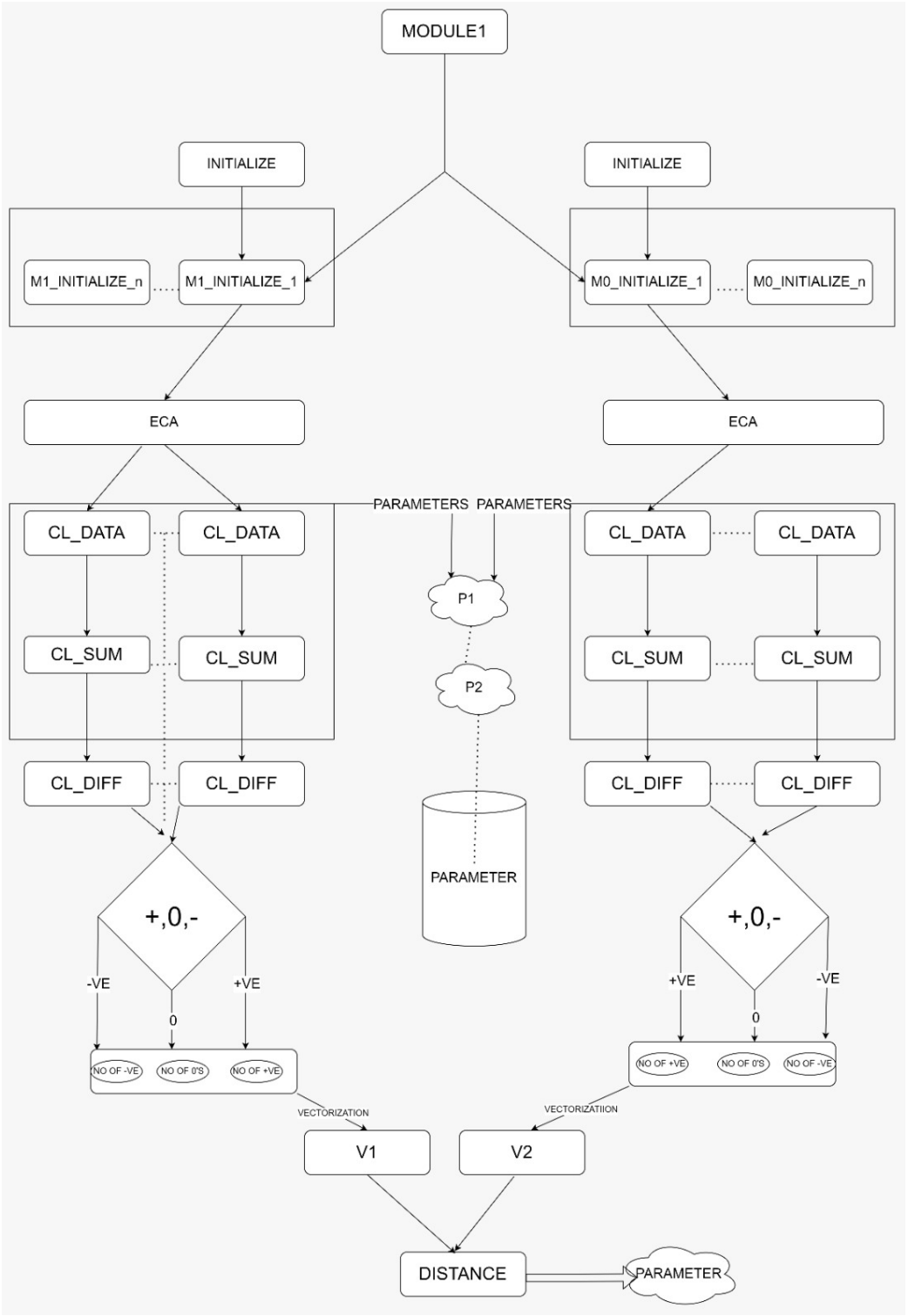

MODULE1
INITIALIZE
INITIALIZE
M1_INITIALIZE_n
M1_INITIALIZE_1
M0_INITIALIZE_1
M0_INITIALIZE_n
ECA
ECA
PARAMETERS PARAMETERS
CL_DATA
CL_DATA
CL_DATA
CL_DATA
P1
CL_SUM
CL_SUM
CL_SUM
CL_SUM
P2
CL_DIFF
CL_DIFF
CL_DIFF
CL_DIFF
PARAMETER
+,0,-
+,0,-
-VE
0
+VE
+VE
0
-VE
NO OF -VE
NO OF 0'S
NO OF +VE
NO OF +VE
NO OF 0'S
NO OF -VE
VECTORIZATION
VECTORIZATIION
V1
V2
DISTANCE
PARAMETER

Figure 4.10: Module-II

4.4.2 Pre-Processing

The preprocessing step is crucial in ensuring the accuracy and effectiveness of any model, as it removes irrelevant information and standardizes the text data, making it easier to extract meaningful insights. It involves a series of steps to clean and prepare the data for further analysis, such as removing punctuation marks,HTML tags.

4.4.3 Parameter extraction

Feature extraction and selection refers to the process of choosing the most important and relevant features from a large set of features to enhance the performance of a machine learning model. The process involves reducing the original set of features into a smaller, more informative subset that can better represent the patterns in the data. The process may use statistical methods, domain knowledge, and machine learning algorithms to identify the most informative and relevant features that can improve the performance of the model while reducing the complexity of the feature space.

- **Number of zeros** : The number of zeros in the cl_sum differences that were calculated by subtracting the original cl_sum and the cl_sum of mutations
- **Number of positive values**: The number of positive values in the cl_sum differences that were calculated by subtracting the original cl_sum and the cl_sum of mutations

- **Number of negative values**: The number of negative values in the cl_sum differences that were calculated by subtracting the original cl_sum and the cl_sum of mutations
- **List of negative values**: Making a list of all negative values in the cl_sum differences that were calculated by subtracting the original cl_sum and the cl_sum of mutations
- **List of positive values**: Making a list of all positive values in the cl_sum differences that were calculated by subtracting the original cl_sum and the cl_sum of mutations
- **Positive minimum value**: The minimum positive value in the cl_sum differences that were calculated by subtracting the original cl_sum and the cl_sum of mutations
- **Positive maximum value**: The maximum positive value in the cl_sum differences that were calculated by subtracting the original cl_sum and the cl_sum of mutations
- **Negative minimum value**: The negative maximum positive value in the cl_sum differences that were calculated by subtracting the original cl_sum and the cl_sum of mutations
- **Negative maximum value**: The positive maximum positive value in the cl_sum differences that were calculated by subtracting the original cl_sum and the cl_sum of mutations

- **Vector**: The cl differences of every mutations are categorized into positive, negative, and zero. The number of positive, negative, and zero differences are counted and used to create a vector with coordinates representing the counts for each category. This process is repeated for both mutations 0 to 1 and 1 to 0, resulting in two vectors. The distance between the two vectors is then calculated.The combination of the number of zeros, number of positive values, and number of negative values is called a vector(v=[no of positive,no of negative,no of zero]).

- **From zeros and ones mutations**:

  - **Vector distance**: It refers to the Euclidean distance between the two vectors that are obtained in the two mutations, i.e., 0 to 1 and 1 to 0.

  - **Euclidean distance**: It is a measure of the distance between two points in a multi-dimensional space. In this case, it is the distance between the vectors obtained in the two mutations.

- **From original cl_data**:

  - **Cl sum greater than threshold**: It refers to finding the noise in the cl data by plotting the data and setting a threshold, and then calculating the sum of the data points that are above this threshold.

  - **Threshold**: It is a value that is set based on the plotted data to distinguish between the noise and the actual data.

  - **Maximum original cl data**: It refers to the maximum cl value in the original cl data that is obtained without any mutations.

Algorithm 2 Mutation Based algorithm

1: **Input** : *Text*

2: **Output** : *sentiment of Data*

3: *Pre-process each sentence in the dataset for generating rule vectors.*

4: *Convert pre-processed data into CA rule vectors using rule values.*

5: *Find out the CL sequences using CA rule vectors.*

6: *Find NMCL locations from the original CL sequences.*

7: **zero to one mutation:**

- Initialize the cellular automata sequence with alternate ones and zeros, and replace the NMCL location with all ones.

8: **one to zero mutation:**

- Initialize the cellular automata sequence with alternate ones and zeros, and replace the NMCL location with all zeros.

9: *Find the CL data for the mutations.*

10: *Find the Cl sum for each mutated CL sequence.*

11: *Find the CL difference for each mutated CL sequence.*

12: *Find the parameters from CL differences.*

- Original Maximum cycle length values(MCL).
- Sum of original Cycle length values.
- Cycle length differences.
- Vector distance.

13: *Implement a sequential LSTM model for sentiment classification.*

14: *Use adam optimiser and Relu or sigmoid activation function.*

15: *The performance analysis is accuracy using the following formula:*

$$Accuracy = \frac{TotalPositive + TotalNegative}{\text{total \# of sentences in data}} \tag{4.3}$$

Chapter 5

Experimental Results

The following section details the experimental methodology employed, which includes the configuration of different models and the testing of multiple datasets. It also offers an outline of the results obtained from these experiments. In essence, the section describes the approach taken to test and evaluate the performance of the models under different conditions and datasets, as well as the outcomes obtained from these tests.

5.1 Experimental Settings

Our main strategies for multilingual sentiment classification included a rule-based approach and a mutation-based approach. In the mutation-based approach, we used both LSTM(Long Short Term Memory) and SVM(Support Vector Machine) algorithms to make predictions. The rule-based approach uses prediction parameters, such as CSI, Euclidean distance, and R-values, to make predictions. These parameters are defined based on prior knowledge of the problem and are used to establish rules that govern how sentiment is classified. By using these parameters, the system can assign sentiment labels to new data points based on their similarity to existing data points with known sentiment labels.

5.2 Results

The Results obtained by using the Mutations approach are are shown in the Table 5.1.

DATASET	METHOD	ACCURACY
English(Twitter)	Mutation(SVM)	70.73
English(Twitter)	Mutation(LSTM)	94.06
English(Twitter)	Mutation(Xgboost)	70.73
English(Reviews)	Mutation(SVM)	54.98
English(Reviews)	Mutation(LSTM)	53.66
Bengali	Mutation(SVM)	74.9
Bengali	Mutation(LSTM)	95.4
Bengali	Mutation(Bi-LSTM)	75.8
Bengali	Mutation(Xgboost)	72.38
Hindi	Mutation(SVM)	83.2
Hindi	Mutation(LSTM)	82.6

Table 5.1: Mutation Based Results

The Results obtained by using the Rule Based approach are are shown in the Table 5.2.

DATASET	METHOD	ACCURACY
Bengali	Rule Based	86.65
English(Reviews)	Rule Based	53.37
English(Twitter)	Rule Based	92.60
Hindi	Rule Based	81.90
Russian	Rule Based	52.45

Table 5.2: Rule Based Results

DATASET	METHOD	ACCURACY
Bengali	LSTM	75.13
French	LSTM	60.5
English(Reviews)	LSTM	51.6
English(Twitter)	LSTM	48.5
Hindi	LSTM	81.0
Russian	LSTM	55.7

Table 5.3: Direct Results

5.3 Comparision with other models

In our study, we assess the effectiveness of our model by comparing its accuracy with that of other models.

Author	DATASET	METHOD	ACCURACY
Samina[12]	Bengali	BERT	78.94
Partha Chakraborty, et.al [3]	Bengali	LSTM(TF-IDF)	96.95
Proposed model(Rule based)	Bengali	Rule based	86.65
Proposed model(Mutations)	Bengali	LSTM	95.4

Table 5.4: Bengali-Language Comparision

Author	DATASET	METHOD	ACCURACY
Santosh et al[6]	Hindi	CNN and Word embeddings	85
Proposed model(Rule based)	Hindi	Rule based	78.9
Proposed model(Mutation based)	Hindi	SVM	83.2

Table 5.5: Hindi-Language Comparision

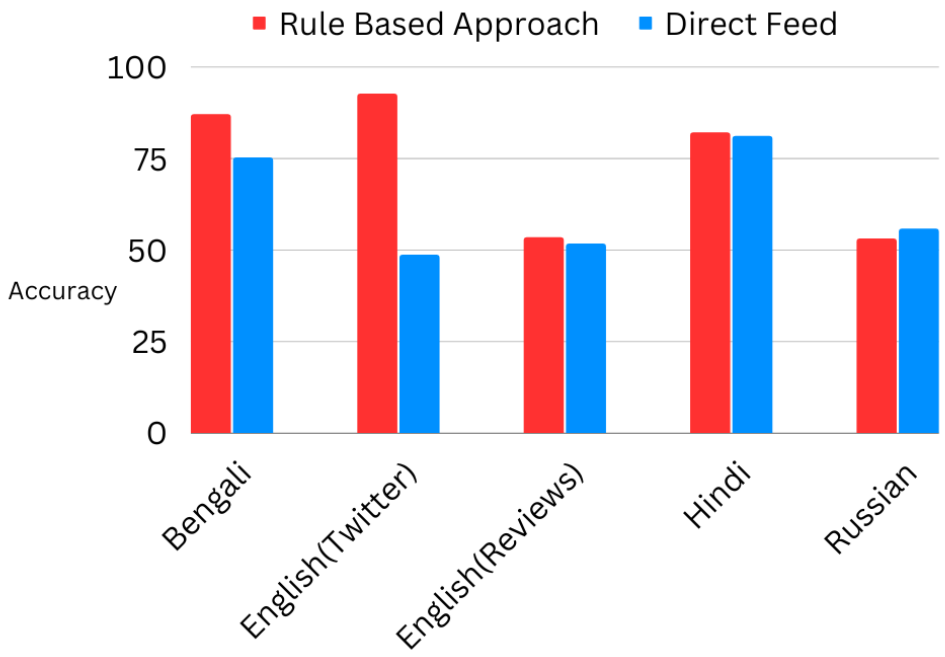


Figure 5.1: Comparision-1

Author	DATASET	METHOD	ACCURACY
Elizabeth M J et al [4]	English	CA Based model	92.44
Proposed model(Rule based)	English	Rule based	92.6
Proposed model(Mutation based)	English	LSTM	94.06

Table 5.6: English-Language Comparision

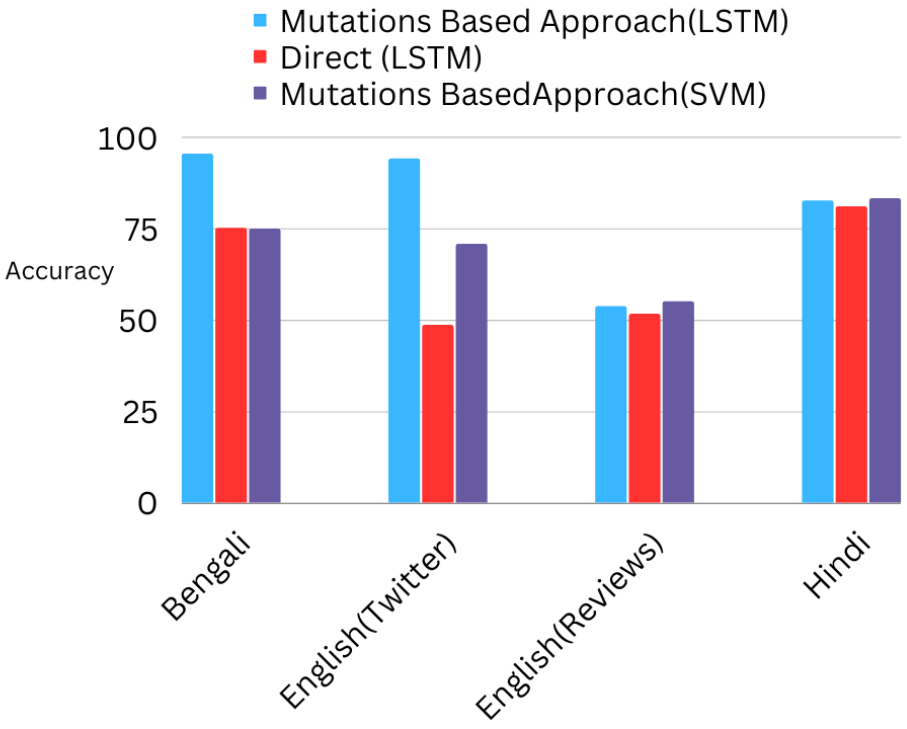


Figure 5.2: Comparision-2

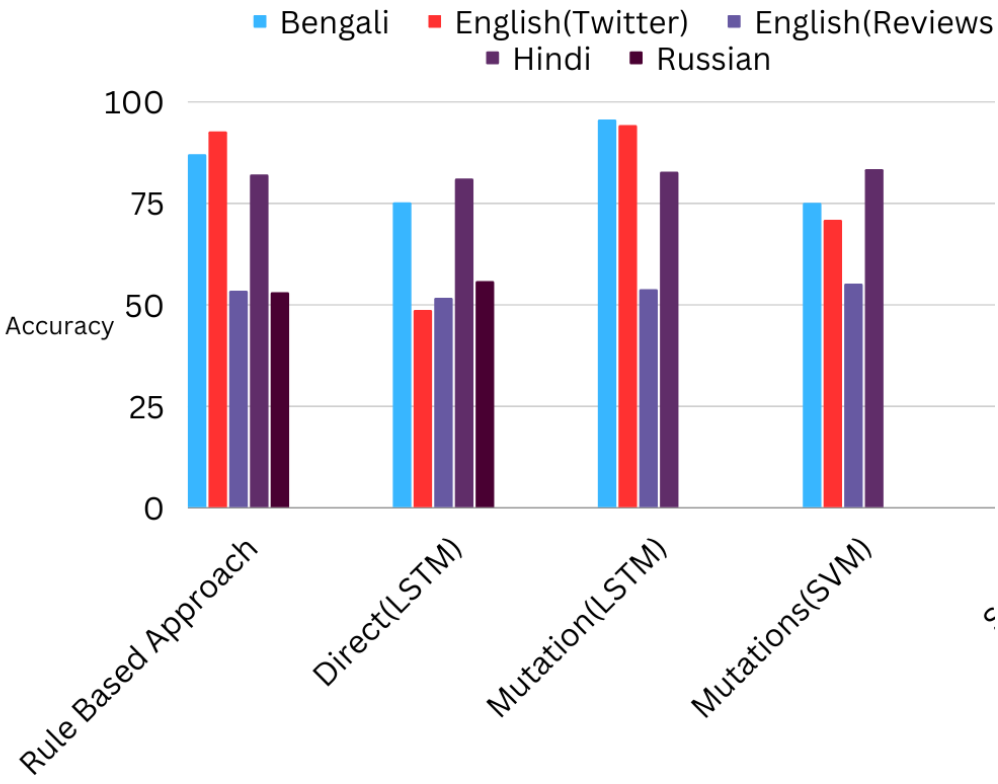


Figure 5.3: Comparision-3

Chapter 6

Conclusion

The focus of this work is on presenting a novel approach for multilingual sentiment analysis by utilizing cellular automata signals. The proposed model does not rely on pre-trained word embeddings, making it a unique and innovative solution for multilingual sentiment analysis. As previously stated, the model underwent training using various datasets, and the outcomes of the different models were recorded and compared in our work . After examining the results of our proposed models and comparing them with those of other models, it can be concluded that the mutations-based approach and Rule base approach can be used for multilingual sentiment analysis.

Bibliography

[1] R. Alroobaea. Sentiment analysis on amazon product reviews using the recurrent neural network (rnn). *International Journal of Advanced Computer Science and Applications*, 13(4), 2022.

[2] R. Bardhan, M. Sunikka-Blank, and A. N. Haque. Sentiment analysis as tool for gender mainstreaming in slum rehabilitation housing management in mumbai, india. *Habitat International*, 92:102040, 2019.

[3] P. Chakraborty, F. Nawar, and H. A. Chowdhury. Sentiment analysis of bengali facebook data using classical and deep learning approaches. In *Innovation in Electrical Power Engineering, Communication, and Computing Technology: Proceedings of Second IEPCCT 2021*, pages 209–218. Springer, 2022.

[4] M. Elizabeth, S. M. Parsotambhai, and R. Hazari. Cellular automata enhanced machine learning model for toxic text classification. In *International Conference on Cellular Automata for Research and Industry*, pages 346–355. Springer, 2022.

[5] M. J. Elizabeth, S. M. Parsotambhai, and R. Hazari. Cellular automata enhanced machine learning model for toxic text classification. In B. Chopard, S. Bandini, A. Dennunzio, and M. A. Haddad, editors, *Cellular Automata - 15th International Conference on Cellular Automata for Research and Industry, ACRI 2022, Geneva, Switzerland, September*

12-15, 2022, Proceedings, volume 13402 of *Lecture Notes in Computer Science*, pages 346–355. Springer, 2022.

[6] V. Gupta, N. Jain, S. Shubham, A. Madan, A. Chaudhary, and Q. Xin. Toward integrated cnn-based sentiment analysis of tweets for scarce-resource language—hindi. *Transactions on Asian and Low-Resource Language Information Processing*, 20(5):1–23, 2021.

[7] R. Hazari and P. P. Chaudhuri. Analysis of coronavirus envelope protein with cellular automata (CA) model. *CoRR*, abs/2202.11752, 2022.

[8] A. Joshi, A. Balamurali, P. Bhattacharyya, et al. A fall-back strategy for sentiment analysis in hindi: a case study. *Proceedings of the 8th ICON*, 2010.

[9] N. Mukhtar and M. A. Khan. Urdu sentiment analysis using supervised machine learning approach. *International Journal of Pattern Recognition and Artificial Intelligence*, 32(02):1851001, 2018.

[10] P. Ray and A. Chakrabarti. A mixed approach of deep learning method and rule-based method to improve aspect level sentiment analysis. *Applied Computing and Informatics*, 2020.

[11] S. E. Saad and J. Yang. Twitter sentiment analysis based on ordinal regression. *IEEE Access*, 7:163677–163685, 2019.

[12] M. M. Samia, A. Rajee, M. R. Hasan, M. O. Faruq, and P. C. Paul. Aspect-based sentiment analysis for bengali text using bidirectional encoder representations from transformers (bert).

[13] T. Thongtan and T. Phienthrakul. Sentiment classification using document embeddings trained with cosine similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, Florence, Italy, jul 2019. Association for Computational Linguistics.

[14] S. Vashishtha and S. Susan. Fuzzy rule based unsupervised sentiment analysis from social media posts. *Expert Systems with Applications*, 138:112834, 2019.

[15] F. Xu, Z. Pan, and R. Xia. E-commerce product review sentiment classification based on a naïve bayes continuous learning framework. *Information Processing & Management*, 57(5):102221, 2020.