JADAVPUR UNIVERSITY

Bengali Text Classification Using SVM

BY

Surajit Maity

EXAMINATION ROLL NO. : MCA2360012

UNDER THE SUPERVISION OF

PROF.(DR.) KAMAL SARKAR

PROJECT SUBMMITION OF FULFILLMENT FOR THE DEGREE OF MASTER OF COMPUTER APPLICATION

IN THE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
FACULTY OF ENGINEERING AND TECHNOLOGY

2023

Certificate of Recommendation

This is to certify that the thesis entitled " Bengali Text Classification Using SVM" has been carried out by Surajit Maity (University Roll Number: 002010503002, Examination Roll No: MCA2360012, University Registration Number: 154210 (2020-2021)), under the guidance and supervision of Prof. (Dr.) Kamal Sarkar, Department of Computer Science and Technology, Jadavpur University, Kolkata, is being presented for the partial fulfilment of the Degree of Master of Computer Applications during the academic year 2022-2023. The research results presented in the thesis have not been included in any other paper submitted for the award of any degree in any other university or institute.

Prof. (Dr.) Kamal Sarkar (Thesis Supervisor)

Department of Computer Science and Engineering

Jadavpur University, Kolkata-32

Countersigned

Prof. (Dr.) Nandini Mukhopadhyay

Head of Department

Computer Science and Engineering

Jadavpur University, Kolkata-32

Prof. Ardhendu Ghoshal

Dean

Faculty of Engineering and Technology

Jadavpur University, Kolkata-32

Jadavpur University
Faculty of Engineering and Technology
Department of Computer Science and Engineering

CERTIFICATE OF APPROVAL

This is to clarify that the project entitled "**Bengali Text Classification Using SVM**" has been completed by Surajit Maity**.** This work is applied under the supervision of **Prof.(Dr.) Kamal Sarkar** in partial fulfilment for the award of the degree of **Master of Computer Applications** of the **Department of Computer Science and Engineering**, **Jadavpur University**, during the academic year **2022-2023**. The project report has been approved because it satisfies the tutorial requirements in respect of project work prescribed for the said degree.

……

Signature of Examiner 1

Date:

……

Signature of Examiner 2

Date:

Date:

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis entitled " Bengali Text Classification Using SVM" contains original research work by the undersigned candidate, as part of his degree of Master of Computer Applications.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name : Surajit Maity

Class Roll No. : 002010503002

Examination Roll No. : MCA2360012

University Registration No. : 154210 (2020-2021)

ACKNOWLEDGMENT

I am delighted to convey my gratitude to my thesis supervisor Prof. (Dr.) Kamal Sarkar , Department of Computer Science & Engineering, my supervisor, for his overwhelming support and encouragement towards accomplishment throughout the duration of the project without which this work would not have been possible. His positive outlook and confidence inspired me and gave me confidence. I am grateful for his support, encouragement, and his valuable suggestions to complete this work. I feel deeply honored that I got this opportunity to work under him.

I would like to express my sincere thanks to all my teachers for providing a sound knowledge base and cooperation.

I would like to thank all the faculty members of the Department of Computer Science & Engineering of Jadavpur University for their continuous support. Last, but not the least, I would like to thank my batch mates for staying by my side when I need them.

_____

Surajit Maity

Roll No: 002010503002

Reg. No. : 154210 (2020-2021)

Examination Roll No. : MCA2360012

CONTENTS

Abstract

In current scenario, lot of online news is available for different topics on Internet from which textual data is increasing rapidly. Due to this, text classification becomes essential to organize them properly so that important news can be searched easily as well as to avoid data loss. One effective solution for this problem is to classify the news into different classes or to extract most important and useful information. With the rapid growth of Text sentiment analysis, the demand for automatic classification of electronic documents has increased by leaps and bound. The paradigm of text classification or text mining has been the subject of many research works in recent time. In this paper we propose techniques for Bengali text classification. In one method we take all the words from document and in another method we extract keywords from each document. We found the Support vector machine (SVM) is the most appropriate to work with our proposed model. The achieved results show significant increase in accuracy compared to earlier methods.

Chapter 1

Introduction

The classification process in text classification involves assigning predefined categories or classes to text documents based on their content. This process is similar to function mapping in mathematics, where the goal is to map inputs to specific outputs.

In text classification, the texts to be classified are represented as a set D, where D = {doc_1, doc_2, doc_3, ..., doc_m}, with m documents in total. Additionally, there is a set of predefined classes or categories C, where C = {cls_1, cls_2, cls_3, ..., cls_n}, with n classes.
The classification process can be interpreted as mapping each document in set D to a specific class from set C. The goal is to accurately assign the most appropriate class to each document based on its content and characteristics. So the classification process can be interpreted as [1]

$$F : D \rightarrow C$$

It's worth mentioning that while your focus is on single-label classification, where each text is assigned to a single class, there are also techniques for multi-label classification, where texts can belong to multiple classes simultaneously. Multi-label classification [2] is used when texts can have multiple topics or attributes associated with them. Which will not be discussed here. In this study, all text is mapped to a single class.

To perform text classification, supervised machine learning methods are commonly used. These methods involve training a model on a labeled dataset, where each document is already assigned to a specific class. The model learns patterns and features from the labeled data and uses them to make predictions on new, unseen documents.

During the training phase, the model analyzes the textual content of the documents and extracts relevant features. These features can include word frequencies, word distributions, or more advanced linguistic features. The model then builds a representation of the documents and their corresponding classes.

Once the model is trained, it can be used to classify new, unlabeled documents. The model applies the learned patterns and features to the unseen text and predicts the most appropriate class based on its knowledge of the training data.

Text classification has various applications, including sentiment analysis, spam filtering, topic categorization, and document classification in many domains such as news articles, customer reviews, and social media posts.

The increasing amount of text information in various fields of life due to the rapid development of Internet and information technology, particularly in the era of big data. Manual text classification, which was previously employed to categorize information, is now deemed inadequate due to its time-consuming, labor-intensive, and costly nature. As a result, automatic text classification has emerged as a solution to efficiently and accurately summarize text from vast amounts of information. This automated approach has gained significant attention in both academic and industrial sectors, becoming a topic of considerable discussion.

Natural Language Processing (NLP) has gained prominence in recent years, leading to the development of numerous tools and techniques for text comprehension. One such tool is the Natural Language Processing Toolkit (NLTK) [3], which is specifically designed to statistically understand and process text data.
The NLTK offers several applications for text analysis, including:

a. Word tokenization: This involves breaking down a sentence into individual words, enabling further analysis at the word level.

b. Sentence tokenization: It entails splitting a paragraph into separate sentences, facilitating sentence-level analysis.

c. Word count: This application involves determining the frequency of word occurrences in a particular document or corpus, providing insights into word usage.

d. Stemming: This technique aims to reduce words to their root form, allowing for more effective processing in natural language

applications.

e. Stop-word removal: Stop-words refer to common words in the English language, such as prepositions and conjunctions. Many text processing applications eliminate these stop-words as they typically carry little relevant information.

By utilizing these NLTK applications, text processing tasks become more efficient and enable the extraction of meaningful insights from text data.

Now-a-days the amount of information available on the web is tremendous and increasing at an exponential rate. To effectively manage this vast volume of data, automatic text classification has emerged as a crucial application and research topic. Since the advent of digital documents, automatic text classification has been employed to categorize and organize the abundance of web-based information.

Automatic text classification relies on machine learning techniques to develop classifiers that learn the distinguishing characteristics of different categories from a set of pre-classified documents. It plays a significant role in various tasks, including information extraction, summarization, text retrieval, and question-answering.

The data used for classification is typically heterogeneous, sourced from various platforms such as web pages, newsgroups, bulletin boards, as well as broadcast or printed news articles, scientific papers, movie reviews, and advertisements. These sources introduce diversity in formats, preferred vocabularies, and writing styles, even within a single genre. Consequently, automatic text classification becomes crucial for efficiently organizing and extracting information from these diverse sources.

Chapter 2

Literature Survey

Natural Language Processing (NLP) applications are commonly used for text processing, classification, and clustering tasks. One of the simplest text analytics applications is spam filtering, which is often employed in email systems. Spam filtering involves classifying incoming emails as either spam or legitimate based on the content within the email. Machine learning techniques are commonly utilized in spam filters to automatically classify emails as spam or non-spam based on the text present in the email. This application demonstrates how NLP can be used to automatically categorize and filter text data to enhance email management and reduce unwanted spam messages.
Different text processing techniques, are as follows.

2.1 Bag of Words

Bag of words [4] is a technique where the sentence is represented by the occurrence or absence of a word in the given sentence. The sentence "টানা ব্যর্থ কেএল রাহুল" Each column represents a unique word from the sentence, and each row represents the binary occurrence (1) or absence (0) of that word in the sentence.

রবিবার	টানা	কেএল	ম্যাচে	সাফল্য	ব্যর্থ	বড়	গাড়ি	নিয়ন্ত্রণ	রাহুল
0	1	1	0	0	1	0	0	0	1

To address the disadvantages of the bag of words technique mentioned:

Loss of Meaning: Bag of words representation treats each word independently and disregards the order and context of the words. This can lead to a loss of meaning. For example, the sentences "Dog bites Man" and "Man bites Dog" would have the same bag of words representation, even though the meaning is different.
Ignoring Word Frequency: Bag of words representation does not consider the frequency of words in the document. It only indicates the presence or absence of a word. As a result, important information about the frequency or importance of certain words is not retained.

These limitations can be addressed by using more advanced techniques, such as n-grams or word embeddings, which consider the context and sequence of words to capture meaning more effectively.

2.2 TF-IDF

TF-IDF, which stands for Term Frequency-Inverse Document Frequency [5], is a commonly used technique in natural language processing and information retrieval to overcome some of the limitations of the bag-of-words approach.

The bag-of-words model represents a text document as a collection of individual words, disregarding their order and context. It considers each word independently, assigning equal importance to all words in the document. This approach has limitations because it does not capture the semantic or contextual information of the words.

TF-IDF addresses this limitation by taking into account both the frequency of a word in a particular document (term frequency) and its rarity across all documents (inverse document frequency).

The term frequency (TF) component of TF-IDF calculates the frequency of a word in a specific document. It gives a higher weight to words that appear more frequently in the document, assuming that such words are more important.

The inverse document frequency (IDF) component measures the rarity of a word across all documents in the dataset. It assigns a higher weight to words that occur less frequently in other documents, assuming that rare words carry more significant meaning.

By combining the TF and IDF components, TF-IDF assigns higher weights to words that are both frequent in a document and rare across other documents. This approach helps to identify important words that are distinctive to a particular document and can provide valuable insights.

2.3 Machine Learning Algorithms

Machine Learning algorithms are used to perform classification, regression, and forecasting tasks effectively. Machine learning algorithms are employed to accomplish tasks such as classification, regression, and forecasting, highlighting their effectiveness in these areas.

The algorithms used for classification tasks are Logistic Regression for binary classification, Decision Trees, Random Forest Classifier, Support Vector Machines, Naive Bayes Classifier, and Boosting Algorithms. There

are several algorithms that are commonly used for classification tasks, including Logistic Regression (specifically for binary classification), Decision Trees, Random Forest Classifier, Support Vector Machines, Naive Bayes Classifier, and Boosting Algorithms. These algorithms are known for their ability to classify data into different categories or classes.
Due to high sparsity and dimensions of the data, only Logistic Regression, Random Forest, and Naive Bayes classifiers are used in modeling the data. Because of the high sparsity (sparse data) and dimensions (large number of features) of the data, only three classifiers are utilized for modeling the data: Logistic Regression, Random Forest, and Naive Bayes. These algorithms are chosen likely because they can handle high-dimensional and sparse data effectively.

2.3.1 Naive Bayes Classifier

McCallum and Nigam [6] focuses on the Naive Bayes classifier for text classification. Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.
Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

$$p(A|B)=\frac{p(B|A)p(A)}{p(B)}$$

Where A and B are events and P(B) ≠ 0
They applyes Bayes' theorem in following way:

$$p(Y|X)=\frac{p(X|Y)p(Y)}{p(X)}$$

Where, y is class variable and X is a dependent feature vector (of size n) where: $X=(x_1, x_2, x_3, \ldots\ldots\ldots, x_n)$
Now, it's time to put a naive assumption to the Bayes' theorem, which is, independence among the features. If any two events A and B are independent, then, p(A,B)=p(A)p(B)
Hence, we reach to the result:

$$P(y|\ x_1, x_2, \ldots\ldots, x_n) = \frac{p(x_1|y)p(x_2|y)\ldots\ldots p(x_n|y)p(y)}{p(x_1)p(x_2)\ldots\ldots p(x_n)}$$

2.3.2 Neural Networks

The field of machine learning and data science has been greatly influenced by neural networks. These models, which are inspired by the human brain's structure and functioning, have become increasingly powerful with the availability of high computing engines. Neural networks can now consist of millions of parameters, enabling them to learn from vast amounts of data and make accurate predictions.

Prasanna and Rao [7] constructed a TFIDF matrix which given to the artificial neural network from which the neural networks learns. Training a neural network indeed involves two major steps, forward propagation and backward propagation.

Forward Propagation: In this step, the neural network takes a set of input data and computes the corresponding output. Each neuron in the network receives input signals, multiplies them by corresponding weights, sums them up, and applies an activation function to produce an output.

The inputs are multiplied by the weights, and the weighted sum is passed through an activation function. The activation function introduces non-linearity to the network, allowing it to learn complex relationships between inputs and outputs. Common activation functions include sigmoid, hyperbolic tangent (tanh), ReLU (Rectified Linear Unit), and others. The activation function maps the input to a specific range or threshold, which becomes the output of that neuron.

Backward Propagation: After the forward propagation step, the network compares the computed output with the desired output and calculates the error. Backward propagation, also known as backpropagation, aims to adjust the weights of the network to minimize this error.

The error is propagated backward through the network, starting from the output layer and moving towards the input layer. The weights of the neurons are updated based on the calculated error and the gradients of the activation function. This process is performed iteratively using optimization algorithms like gradient descent, which adjusts the weights in the direction that reduces the error.

The gradients are computed using the chain rule of calculus, which calculates the impact of each weight on the overall error. By adjusting the weights based on the gradients, the network gradually learns to make better predictions and minimize the error.

The process of forward and backward propagations is repeated iteratively, with the network adjusting the weights in each iteration to improve its

predictions. The goal is to find the optimal set of weights that minimize the error and make accurate predictions on new, unseen data.

During the feedforward propagation stage in neural network training, each input (xi) indeed receives an input signal.

Input Signals: Each input in the neural network receives a specific input signal (xi) associated with the input pattern or data point. These input signals represent the features or attributes of the input data.

Hidden Units: The input signal from each input is then sent to each hidden unit (z1, z2, z3, ...). A hidden unit, also known as a neuron or node, is responsible for processing the input signals and generating an activation signal.

Activation Calculation: The hidden layer calculates its activation signal based on the received input signals. This calculation typically involves summing up the weighted input signals and applying an activation function to the result. The weights represent the connections between the inputs and the hidden units.

Output Unit: Once the activation signal is calculated for each hidden unit, it is sent to the output unit. The output unit uses the received activation signal(s) to compute the output or response for the given input pattern.

Output Comparison: After generating the output value, each output unit compares its activation value (y_k) with the target value (t_k). The target value represents the desired or expected output for the given input pattern. The output unit checks the activation value against the target value to determine the level of error or mismatch.

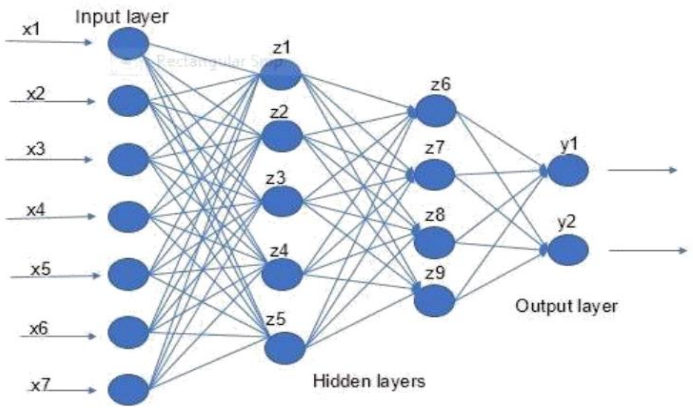

Figure 1: Neural Network Architecture

Chapter 3

Dataset

Our dataset is based on Bengali news. In recent years, news delivery websites have become a popular medium for individuals to access current events information. In our research the Bengali news have been gathered from the online news website ‘Ei Samay’. Dataset contains almost 1756 news. News in these dataset belong to 38 different topics (labels). Each news record consists of several attributes from which we are using only ‘Category’ and ‘Article’.

Chapter 4

Proposed Methodology

Our proposed Bengali text classification using SVM includes several methods: (1) Bengali Text Classification without keywords, (2) Bengali Text classification with keywords, (3) Evaluation and Results.

4.1 Bengali Text Classification without keywords

This method includes several steps: (1) Breaking into sentences, (2) Stop-Word Removal, (3) Document Representation using word embedding, (4) SVM Classifier, (5) Experiments, (6) Final Prediction. A block diagram of the proposed method is shown in Figure 2.

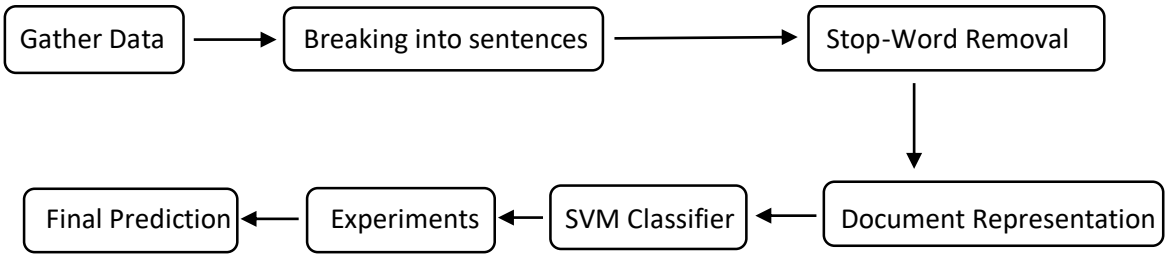

Figure 2: Bengali Text Classification without keywords

4.1.1 Breaking into sentences

Sentence tokenization is performed using the sentence tokenize function of the NLTK library. This process involves breaking the input document set into individual sentences. Sentence tokenization helps in analyzing and processing text at a more granular level, focusing on the meaning and structure of individual sentences.

4.1.2 Stop-Word Removal

Stop words are common words that often appear in a language but typically do not carry significant meaning or contribute to the understanding of the text. Bengali stop word file is used to remove stop words. Removing stop words helps in reducing noise and irrelevant

information, allowing the focus to be on more meaningful words and concepts.

4.1.3 Document Representation

After removing stop words, we get meaningful words for each document. For each word, we search for the corresponding word vector into the list of word vectors obtained from a Bengali pretrained word vector model called Fast Text. Then we calculate the average of the vectors for the words. Thus each document is represented as a vector. Figure 3 shows an example of a vector representing a document taken from our dataset.

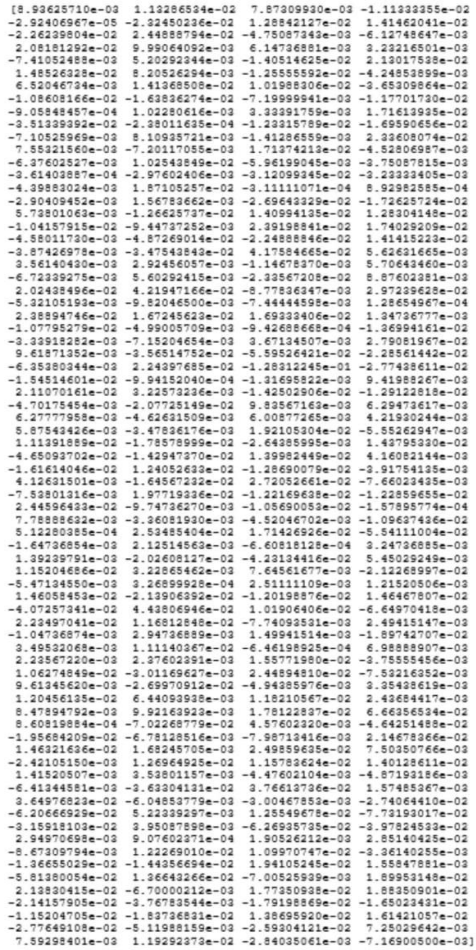

Figure 3: document representation

4.1.4 Classifier

We import the SVM Classification Class from scikit-learn. Support vector machines (SVM) is powerful yet flexible supervised machine learning method used for classification. SVM seeks a decision surface to separate the training data points into two classes and makes decisions based on the

support vectors that are selected as the only effective elements in the training set. Support Vector Machine (SVM) have several advantages that make them stand out compared to other algorithms in certain scenarios. Here are some reasons why SVMs are often considered favourable:

Effective in high-dimensional spaces: SVM perform well in cases where the number of features is greater than the number of samples. This makes them suitable for tasks involving text classification or image recognition, where the dimensionality of the data is typically high.

Robust against overfitting: SVM aim to find the maximum margin, which promotes generalization and helps avoid overfitting. This means that SVM can handle noisy or overlapping data and still make accurate predictions on unseen data.

Versatility with kernel functions: SVM can handle non-linearly separable data by utilizing different kernel functions. The ability to transform the data into a higher-dimensional feature space allows SVM to find complex decision boundaries, making them capable of capturing intricate relationships in the data. The kernel used here is the “rbf” kernel which stands for Radial Basis Function.

Global optimization: The training of SVM involves solving a convex optimization problem, which means they can find the global optimum rather than getting stuck in local optima. This property contributes to their robustness and ensures that SVM consistently converge to a good solution.

Margin-based decision making: SVM focus on maximizing the margin between different classes. This approach leads to better generalization performance, as it encourages the model to learn decision boundaries that are less influenced by noisy or irrelevant data points.

Support for sparse data: SVM can handle datasets with a large number of features, even if most of them are zero (sparse data). This makes SVMs suitable for tasks involving text data or other high-dimensional sparse representations.

4.1.5 Experiments

News in these dataset belong to 37 different topics (labels). Each news

record consists of several attributes from which we are using only 'Category' and 'Article'.
The details of the dataset are available in the table 1.

Class	Number of Articles
Agriculture	50
Business	50
Health	50
Labor_and_Employment	50
Law	50
Miscellaneous	50
Music	50
Politics	50
Public_lands_and_water_management	24
Religion	50
Science	50
Social_welfare	11
Space	46
Sports_other_than_football_and_cricket	50
Caste	42
Technology	50
Transportation	44
Travel	50
Weather	50
World_and_international	50
Cinema	50
Computer	50
Cricket	50
Crime	50
Defence	50
Economy	50
Education	50
Banking	50
Election	50
Electronics	50
Energy	35
Entertainment	50
Environment	50
Family issues	50
Finance	50
Football	50
Government_Operations	50

Table 1: dataset description

We split the data into the training and the test set. 20% of the data kept as the test set and the remaining 80% used for training SVM model.

The training data is scaled, and its scaling parameters are determined by applying a fit_transform() to the training data. We do not want to be biased with our model. We want our test data to be a completely new and a surprise set for our model. The transform method helps us in this case.

Feature scaling is an additional step that can increase the speed of the program as we scale down the values of X to a smaller range. In this, we scale down both the train and the test to a small range of -2 to +2.

4.1.5.1 Parameter tuning

Kernel parameters selects the type of hyperplane used to separate the data. Gamma: gamma is a parameter for non-linear hyperplanes. The higher the gamma value it tries to exactly fit the training data set. C: C is the penalty parameter of the error term. It controls the trade off between smooth decision boundary and classifying the training points correctly. How accuracy varies for different combination of gamma and C values shown in table 2.

gamma	C	Accuracy(in percentage)
0.001	1	52
0.002	1	67
0.003	1	72
0.001	10	85
0.002	10	89
0.003	10	89
0.001	100	92
0.002	100	92
0.003	**100**	**93**

Table 2: Parameter Tuning

4.2 Bengali Text Classification with keywords

This method includes several steps: (1) Keyword selection, (2) document representation and (3) SVM Classifier, (4) Experiments, (5) Final Prediction.

A block diagram of the proposed method is shown in Figure 4.

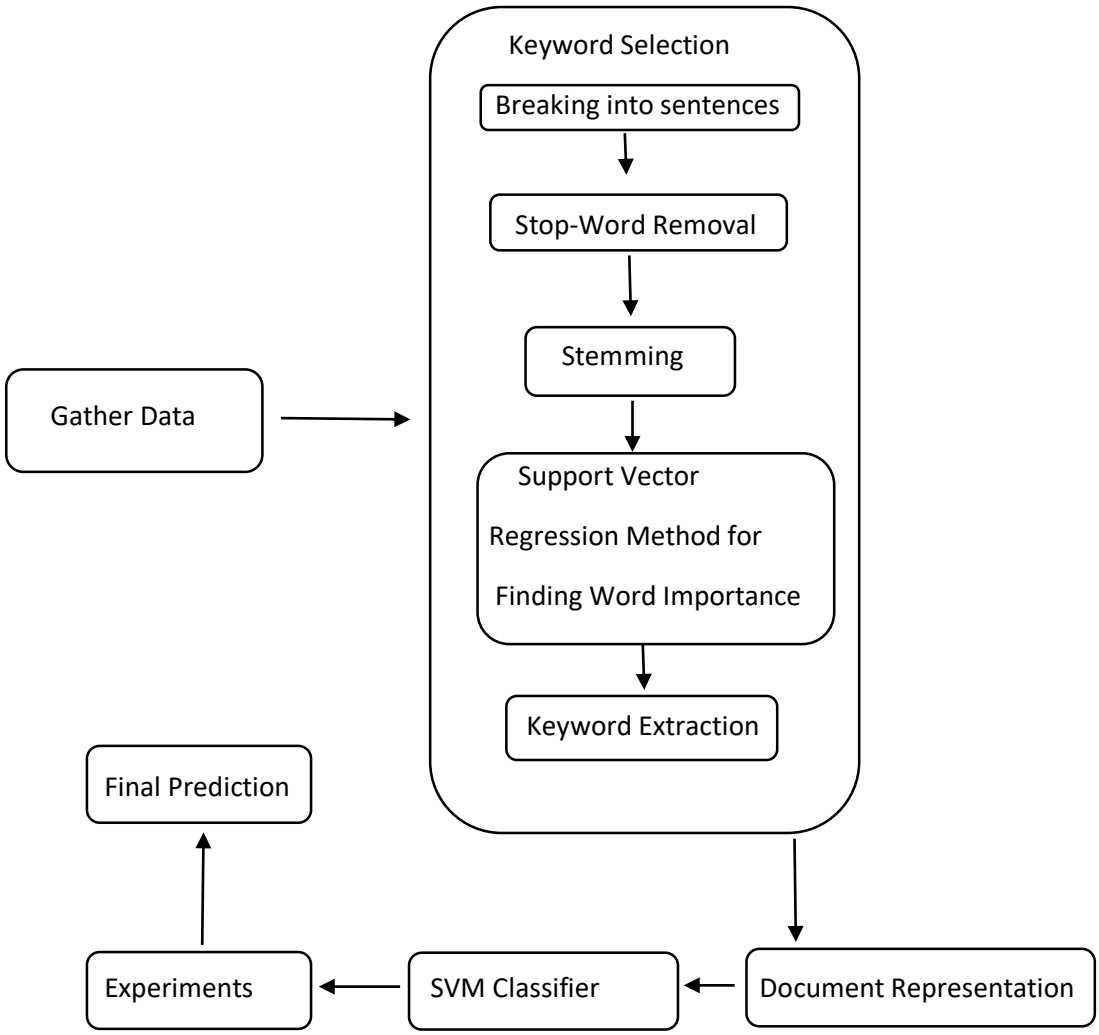

Figure 4: Bengali Text Classification with keywords

4.2.1 Keyword Selection

Chatterjee and Sarkar [8] use a method for predicting word importance (word weight) using the support vector regression model.

4.2.1.1 Pre-processing

This step includes the process:

Sentence Tokenization: Sentence tokenization is performed using the sentence tokenize function of the NLTK library. This process involves breaking the input document set into individual sentences. Sentence

tokenization helps in analyzing and processing text at a more granular level, focusing on the meaning and structure of individual sentences.

Stop Words Removal: Stop words are common words that often appear in a language but typically do not carry significant meaning or contribute to the understanding of the text. Bengali stop word file is used to remove stop words. Removing stop words helps in reducing noise and irrelevant information, allowing the focus to be on more meaningful words and concepts.

Stemming: Stemming is a process that reduces words to their base or root form. It helps in standardizing different word variations to their common base, thereby reducing the dimensionality of the text data. In this step, Bengali steam word file is used. Stemming helps in consolidating related words and capturing their common essence.

4.2.1.2 Word importance prediction using support vector regression model

In the mentioned research, a supervised Support Vector Regression (SVR) model is utilized for predicting the importance of a term. While the frequency of a term is one factor in determining its importance, the research acknowledges that there are other features that contribute to the term's significance. We will discuss various features used for developing the SVR model.

Word position in the document: Since the word importance is dependent on the word's position in the document, we consider this as a feature. The sentences of the document are numbered from 1 to *n*. This feature is computed as follows:

$$\text{POS\_IN\_DOC} = \frac{\text{Position of the sentence in which the word occurs}}{Total\ number\ of\ sentences\ in\ the\ document} \qquad (1)$$

Global Term Frequency (GTF): Since we work with multi-document, frequency of a word w in the entire input collection is considered as a separate feature. The average TF over the number of documents in the input cluster is taken as a feature.

$$\text{GTF(w)} = \frac{1}{|C|} \sum_{w \in \text{C}}^{|C|} \text{TF(w)} \qquad (2)$$

Where TF (w) = total count of occurrences of w in the input collection.

$|C|$= the size of the input collection

TF-IDF Local: Local term frequency (LTF*)* of a word is multiplied by its IDF value to define a new feature. The IDF value of the word *w* is computed over a corpus of N documents using equation (3):

$$ID(w) = log\ [0.5 + \frac{N}{DF(w)}] \quad (3)$$

Where N = corpus size in terms of documents.

TF-IDF Global: This feature is defined by the product of GTF (defined in Equation 2) and IDF as follows:

$$TF - IDF\ Globa(w) = GTF(w) * IDF(w) \quad (4)$$

Proper Noun: The proper nouns like organization name, person name, etc. play an important role in terms selection. For this reason, we consider a feature that checks whether a word is a part of a proper noun or not. This is considered a binary feature. If *w* is the part of a proper noun then the feature value = 1, otherwise its value = 0.

Word length (w): Word length is considered as a feature. It is observed that the words that are longer are highly informative. This is also considered a binary feature. If length (w) > =5 then the value of this feature =1, otherwise its value =0

SVR model is used for predicting the degree of importance for each word. For training, the input to the regression model is represented in the form <*x,y*>, where x is a vector of the values of features described above in this section and y is the target value.

While selecting keywords, the redundant words are removed from the keyword because the redundancy affects keyword quality. A word is selected for a keyword if its similarity with the previously selected word is less than a predefined threshold value.

4.2.2 Document Representation

After keyword extraction, we get keywords for each document. For each keyword, we search for the corresponding word vector into the list of word vectors obtained from a Bengali pretrained word vector model called Fast Text. Then we calculate the average of the vectors for the

keywords. Thus each document is represented as a vector. Figure 5 shows an example of a vector representing a document taken from our dataset.

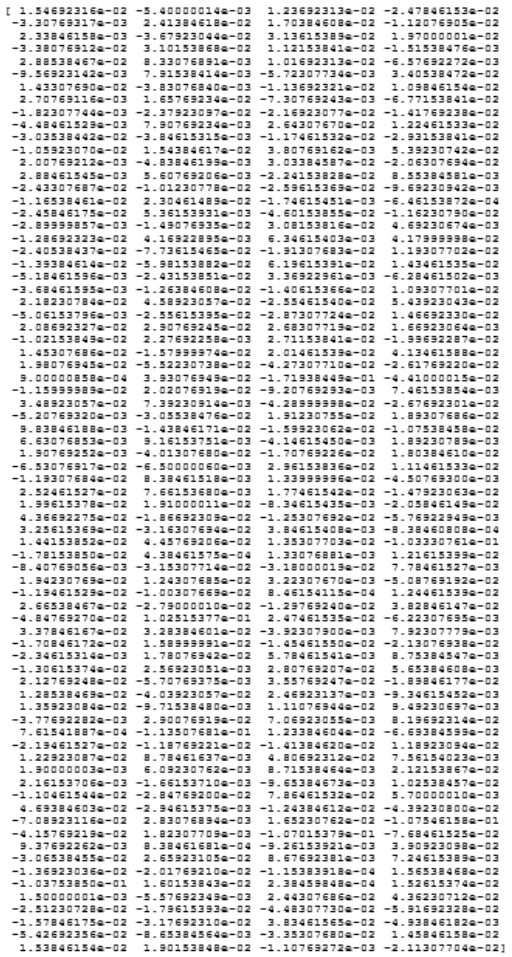

Figure: 5

4.2.3 Classifier

Previously we have discussed in detail about SVM classifier in section 4.1.4. For this experiment we use the same classifier.

4.2.5 Experiments

Detailed dataset description is given in the section 4.1.5. We split the data into the training and the test set. 20% of the data kept as the test set and the remaining 80% used for training SVM model.

The training data is scaled, and its scaling parameters are determined by applying a fit_transform() to the training data. We do not want to be biased with our model. We want our test data to be a

completely new and a surprise set for our model. The transform method helps us in this case.

Feature scaling is an additional step that can increase the speed of the program as we scale down the values of X to a smaller range. In this, we scale down both the train and the test to a small range of -2 to +2.

4.2.5.1 Parameter tuning

Kernel parameters selects the type of hyperplane used to separate the data. Gamma: gamma is a parameter for non-linear hyperplanes. The higher the gamma value it tries to exactly fit the training data set.

C: C is the penalty parameter of the error term. It controls the trade off between smooth decision boundary and classifying the training points correctly.

How accuracy varies for different combination of gamma and C values shown in table 3.

gamma	C	Accuracy(in percentage)
0.001	1	54
0.002	1	64
0.003	1	73
0.001	10	80
0.002	10	82
0.003	10	82
0.001	100	84
0.002	100	84
0.003	**100**	**84**

Table 3: Parameter Tuning

Chapter 5
Evaluation and Results

This section shows the experimental results based on the proposed algorithms. The results for the category classification will be given. The evaluation indicators include: Precision, Recall, f-1 score and Accuracy.

Precision measures the proportion of correctly classified positive instances (documents assigned to a particular class) out of all instances classified as positive. It indicates the ability of the SVM classifier to accurately identify documents belonging to a specific category.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}$$

Recall also known as sensitivity or true positive rate, measures the proportion of correctly classified positive instances out of all actual positive instances. It reflects the ability of the SVM classifier to capture all relevant documents of a particular category

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}$$

Accuracy score that indicates the percentage of correctly classified documents when classified by SVM. Higher accuracy values indicate a more successful classification model.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}$$

F1-score is the harmonic mean of precision and recall. It provides a balanced measure of the classifier's performance, considering both precision and recall. Higher F1-scores indicate better overall performance.

$$\text{F1} - \text{score} = 2\frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

TP: True Positive means that news which is classified to its correct class.

FN: False Negative means that news is classified to a wrong class.

TN: True Negative means that news which does not belong to that class and is misclassified.

FP: false positive is an outcome where the model incorrectly predicts the *positive* class.

Results are obtained from Bengali Text Classification without keywords are shown in table 4 and Bengali Text classification with keywords are shown in table 5.

class	precision	recall	f1-score
Agriculture	1.00	1.00	1.00
Banking	1.00	0.90	0.95
Business	0.91	1.00	0.95
Caste	0.89	1.00	0.94
Cinema	1.00	0.92	0.96
Computer	1.00	1.00	1.00
Crime	1.00	1.00	1.00
Defence	1.00	1.00	1.00
Economy	0.88	1.00	0.93
Education	1.00	0.92	0.96
Electronics	0.75	1.00	0.86
Energy	1.00	0.88	0.93
Entertainment	1.00	1.00	1.00
Environment	1.00	1.00	1.00
Family issues	1.00	1.00	1.00
Finance	1.00	1.00	1.00
Football	1.00	1.00	1.00
Government_Operatios	0.90	1.00	0.95
Health	1.00	0.90	0.95
Labor_and_Employment	1.00	1.00	1.00
Law	1.00	1.00	1.00
Miscellaneous	1.00	1.00	1.00
Music	0.92	1.00	0.96
Politics	0.89	0.89	0.89
Public_lands and_water_managemen t	1.00	0.40	0.57
Science	0.89	1.00	0.94
Space	0.67	1.00	0.80
Sports_other_than_f ootball_and_cricket	1.00	1.00	1.00
Technology	1.00	1.00	1.00
Transportation	1.00	1.00	1.00
Weather	1.00	0.73	0.84
World_and_Internati onal	0.78	1.00	0.88
Cricket	1.00	1.00	1.00
Election	1.00	0.75	0.86
Religion	0.69	1.00	0.81
Travel	0.91	1.00	0.95

Accuracy: 0.94

Table 4: results of Bengali Text Classification without keywords

```
Cross Validation Scores:
[0.96875 0.94886364 0.93465909 0.94034091 0.93181818]
Average CV Score:  0.9448863636363637
Number of CV Scores used in Average:  5
```

class	precision	recall	f1-score
Agriculture	0.92	0.86	0.89
Banking	0.76	0.62	0.68
Business	0.91	1.00	0.95
Caste	1.00	0.36	0.53
Cinema	1.00	1.00	1.00
Computer	1.00	1.00	1.00
Crime	0.92	1.00	0.96
Defence	0.93	1.00	0.96
Economy	0.87	1.00	0.93
Education	0.82	1.00	0.90
Electronics	0.56	1.00	0.71
Energy	0.75	0.60	0.67
Entertainment	0.56	1.00	0.71
Environment	0.85	1.00	0.92
Family issues	0.88	1.00	0.94
Finance	0.89	1.00	0.94
Football	0.86	0.92	0.89
Government_Operatios	0.64	1.00	0.78
Health	0.80	0.89	0.84
Labor_and_Employment	0.88	0.88	0.88
Law	0.91	1.00	0.95
Miscellaneous	0.89	1.00	0.94
Music	1.00	1.00	1.00
Politics	0.93	1.00	0.96
Public_lands and_water_managemen t	0.67	0.67	0.67
Science	0.67	0.86	0.75
Social_welfare	1.00	1.00	1.00
Space	0.83	1.00	0.91
Sports_other_than_f ootball_and_cricket	0.91	0.91	0.91
Technology	0.89	1.00	0.94
Transportation	0.83	0.83	0.83
Weather	0.87	1.00	0.93
World_and_Internati onal	0.85	1.00	0.92
Cricket	0.67	1.00	0.80
Election	1.00	1.00	1.00
Religion	0.70	1.00	0.82
Travel	0.82	1.00	0.90

Accuracy: 0.85

Table 4: results of Bengali Text Classification with keywords

```
Cross Validation Scores:
[0.85795455 0.84090909 0.84659091 0.85511364 0.86647727]
Average CV Score:  0.8534090909090908
Number of CV Scores used in Average:  5
```

Chapter 6

Conclusion and Future Works

With the development of machine learning and deep learning, text classification comes into a new generation, the accuracy is higher and the time consuming is lower. In this thesis, the process of text classification is introduced, extracting keyword from document. Text classification not only can be used in news classification but also in other areas such as spam detection and sentiment analysis. In future these algorithms can be tested on larger corpora. Moreover these algorithms can be improved so that efficiency of categorizations could be improved. A combination of algorithm can be used in order to achieve clustering in a faster way.

References

[1] Li, Saihan. "Text classification Based on Machine Learning Methods." (2019).

[2] Gibaja, Eva, and Sebastián Ventura. "A tutorial on multilabel learning." *ACM Computing Surveys (CSUR)* 47.3 (2015): 1-38.

[3] Loper, Edward, and Steven Bird. "Nltk: The natural language toolkit." *arXiv preprint cs/0205028* (2002).

[4] Vajrala, Ajith. "Text Classification." (2019).

[5] Das, Bijoyan, and Sarit Chakraborty. "An improved text sentiment classification model using TF-IDF and next word negation." arXiv preprint arXiv:1806.06407 (2018).

[6] McCallum, Andrew, and Kamal Nigam. "A comparison of event models for naive bayes text classification." AAAI-98 workshop on learning for text categorization. Vol. 752. No. 1. 1998.

[7] Prasanna, P. Lakshmi, and D. Rajeswara Rao. "Text classification using artificial neural networks." *International Journal of Engineering & Technology* 7.1.1 (2018): 603-606.

[8] Soma Chatterjee & Kamal Sarkar (2022). Predicting Word Importance Using a Support Vector Regression Model for Multi Document Text Summarization. In: International conference on Advance in Data-driven Computing and Intelligent System (ADCIS-2022). (Accepted)