Classification of Comments Supporting the Indian Farmers' Protest Using Active Learning and Weak Supervision

Ajay Biswas

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Classification of Comments Supporting the Indian Farmers' Protest Using Active Learning and Weak Supervision

Thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Technology

in

Computer Science and Engineering
(*Specialization*: *Information Security*)

by

Ajay Biswas

(Roll Number: 220CS2184)

based on research carried out

under the supervision of

Prof. Tapas Kumar Mishra

May, 2022

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Department of Computer Science and Engineering
National Institute of Technology Rourkela

Prof. Tapas Kumar Mishra
Assistant Professor

May 23, 2022

Supervisor's Certificate

This is to certify that the work presented in the thesis entitled *Classification of Comments Supporting the Indian Farmers' Protest Using Active Learning and Weak Supervision* submitted by *Ajay Biswas*, Roll Number 220CS2184, is a record of original research carried out by him under my supervision and guidance in partial fulfillment of the requirements of the degree of *Master of Technology* in *Computer Science and Engineering*. Neither this thesis nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

_____
Tapas Kumar Mishra

Dedication

I dedicate this thesis to my parents. Without their love and support, the completion of this work would not have been possible.

Signature

Declaration of Originality

I, *Ajay Biswas*, Roll Number *220CS2184* hereby declare that this thesis entitled *Classification of Comments Supporting the Indian Farmers' Protest Using Active Learning and Weak Supervision* presents my original work carried out as a postgraduate student of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the thesis. Works of other authors cited in this thesis have been duly acknowledged under the sections "Reference" or "Bibliography". I have also submitted my original research records to the scrutiny committee for evaluation of my thesis.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present thesis.

May 23, 2022
NIT Rourkela

Ajay Biswas

Acknowledgment

This thesis, though an individual work, has benefited in various ways from several people. While it would be simple to name them all, it would not be easy to thank them enough.

The enthusiastic guidance and support of *Prof. Tapas Kumar Mishra* inspired me to stretch beyond my limits. His patience, motivation, and immense knowledge have helped me produce quality research as well as improve my domain knowledge and technical writing skills. My solemnest gratitude to him.

Many thanks to my faculties, fellow research colleagues, friends, and everyone who helped me on this journey. It gives me a sense of joy to be with you all.

Finally, my heartfelt thanks to my family for their unconditional love and support. My words fail me to express my gratitude to my beloved parents, who sacrificed their comfort for my betterment.

May 23, 2022
NIT Rourkela

Ajay Biswas
Roll Number: 220CS2184

Abstract

The introduction of farm bills 2020 was seen as a major agricultural reform. However, started a year-long protest which finally ended with the repeal of the bills. In this work, the authors tried to classify YouTube comments into those that are in support of the farm bill and those that are against it. A total of 1076 unique videos were gathered from YouTube, consisting of a total of 178,608 comments, out of which 20,024 comments were used to form the corpus. The authors proposed a batch-wise random sampling and uncertainty sampling technique that significantly reduced labeling costs by 75% with decent accuracy. For performing classification, four classifiers were incorporated, i.e., Logistic Regression, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Multi-Layer Perceptron (MLP), along with FastText for Word Embeddings. Among these four classifiers, SVM and MLP performed the best, with 82.64% and 82.18% accuracy, respectively. The results of the KNN classifier were further improved to 72.66% after performing Weak Classification combined with Uncertainty Sampling.

Keywords*: *active learning*; *farmer protest*; *fasttext*; *uncertainty sampling*; *weak classification

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The Farm Bills, or the Indian agriculture acts of 2020, are three acts initiated by the Parliament of India in September 2020. The three farm acts are as follows: ”Farmers’ Produce Trade and Commerce (Promotion and Facilitation) Act, 2020; Farmers (Empowerment and Protection) Agreement on Price Assurance and Farm Services Act, 2020; and Essential Commodities (Amendment) Act, 2020”. Although the bills were supposed to be beneficial to the farmers, they led to a mass protest, which gained momentum in September 2020 [1]. The protest continued for over a year and finally was repealed in the month of November, 2021. The protest was not only limited to the grounds, but also spread across various social media websites like YouTube, Reddit, Facebook, Instagram, and Twitter. Our main goal is not to argue who is right or wrong but to identify the comments that are in favor and against the Indian Farmers’ protest.

The proposed approach is inspired by the work done by Palakodety et al.[2]. Their research proposed an Active Learning (AL) based classifier that can categorize comments in favor of the Rohingyas. Further discussions related to this paper will be dealt with in the literature survey section. Active learning is a technique for reducing manual annotation efforts during training phase of machine learning. The annotation is done by a human (called the ”oracle”) which helps AL systems achieve high accuracy with few labelled instances. For problems having a large collection of unlabeled data, pool-based sampling is used [3], as shown in Figure 1.1. AL is highly useful in classifying comments that involves a person’s opinion, belief or political interest, as it’s very tough to label large numbers of comments accurately and effortlessly by a human. Also, there are numerous challenges to be dealt with before any classification can take place. Some of the challenges are (i) dealing with multiple languages having different levels of grammatical accuracy, (ii) unstructured data, (iii) ambiguous sentences, (iv) unrelated comments, etc.

1.2 Applications of Active Learning

Active learning is becoming a burning topic in the field of machine learning due to its high performance and wide range of uses. Some of the areas where active learning can be useful are as follows:

- *Speech Recognition.* One by one labeling speech utterances can be very challenging [4] as well as time consuming. As speech may have several languages or multiple dialects, trained linguists are needed for this purpose.
- *Information Extraction.* To extract high quality information, hours of manual labor is required. For highly specialized job, professional are required like Genomic information retrieval requires Phd-level candidates [5].

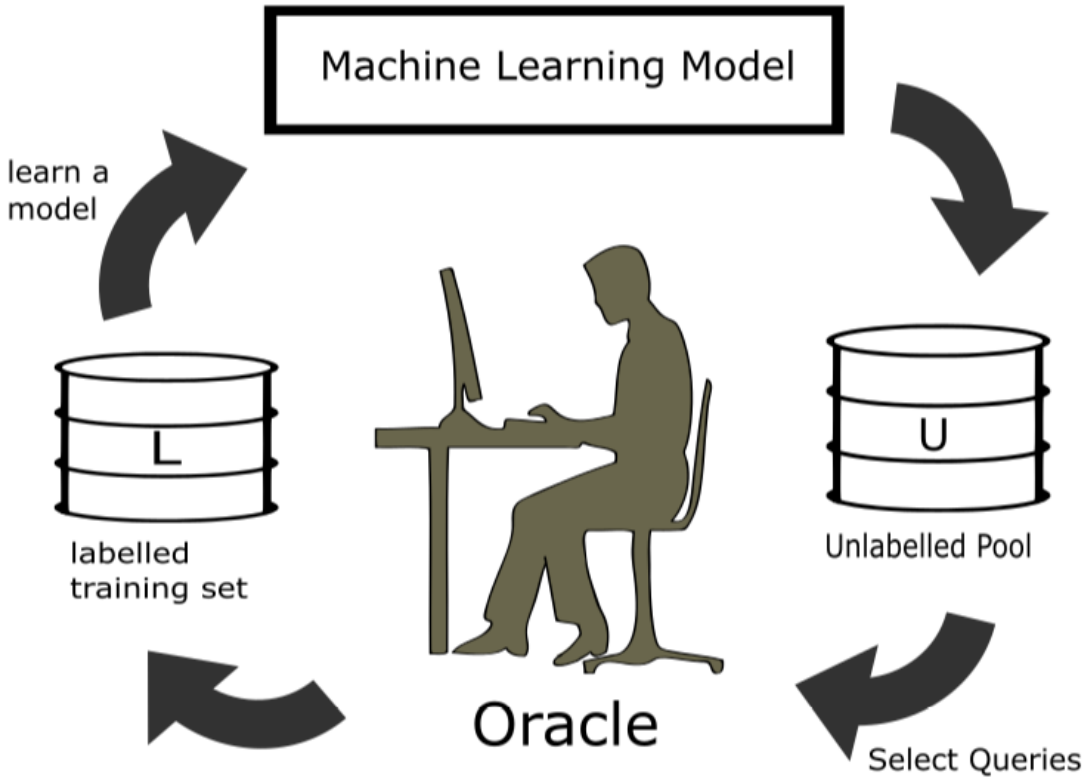


Figure 1.1: Pool-based Active Learning

1.3 Word Embedding Techniques

Humans can read sentences and understand their meaning, but computers don't work this way. The text data has to be transformed into numerical data, which they will use for

classification. These numerical data are known as ”word embeddings”. This is an important step before classification and can give fascinating results if a good technique is used. We will be extensively using two word embedding techniques in our project to generate word embeddings.

1.3.1 TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) is a technique that quantifies words in a set of documents. It is used to measure the importance of words in data. The Term Frequency (TF) is the number of times a word appears in a document divided by the total number of words in that particular document. Inverse Document Frequency (IDF) is the document frequency of a word among all documents. The formula of TF-IDF is shown in Equation 1.1.

$$\begin{aligned} TF(t,d) &= \frac{f_d(t)}{|d|} \\ IDF(t,D) &= \ln \frac{|D|}{|\{d \epsilon D : t \varepsilon d\}|} \\ TF(t,d) - IDF(t,D) &= TF(t,d) \times IDF(t,D) \end{aligned} \tag{1.1}$$

Where $f_d(t)$ is the number of times term t appeared in document d, $|d|$ is the number of words in document d, and $|D|$ is the total number of documents.

1.3.2 FastText

FastText [6] is an open-source library from Facebook for word embeddings and text classification. It uses hierarchical classifiers to build models, which is faster compared to models built through deep neural networks, which uses linear classifiers. FastText provides both supervised and unsupervised learning and supports both Continuous Bag of Words (CBOW) and Skip-gram models [7]. For unsupervised learning, where labels are not available, FastText uses the N-Gram Technique to split words into n-gram components, which capture the meaning of suffixes and prefixes. FastText can also provide embeddings of Out of Vocabulary (OOV) words, which gives it an upper edge against other competitors.

1.4 Motivation

Voting is an important part of democracy as it allows people to choose their representatives. Just like citizens choose their representatives, they also agree or disagree with government policies. By analyzing comments from social media sites, we can understand the sentiment of the country. The motivation behind the selection of Active Learning for comment

classification is its practicality and performance, which is superior to other techniques when it comes to real-life based situations where comments can be repetitive, have a lesser degree of subjective and language proficiency, or too complex to understand.

1.5 Organization of the Thesis

The organization of this thesis is as follows: Chapter 1 provides a brief introduction which describes about the ongoing farmers' protest and how active learning can be used to classify comments supporting the protest. It also describes the work's motivation and goal. Chapter 2 provides a brief summary of the related works,which are summarized in this chapter; and Chapter 3 describes the proposed work. It describes how the dataset was built and how active learning and weak supervision were implemented to make the labelling process faster. Also, it highlights the findings of one similar piece of work done in the field of active learning. Finally, Chapter 4 presents the conclusion of the proposed work, the limitations, and the areas that can be improved in the future.

Chapter 2

Literature Survey

2.1 Overview

This chapter contains a brief survey of research work done in the field of Sentence Classification and Active Learning.

2.2 Related Works

Previously various research were conducted in the field of Active Learning and Sentence Classification. Some of the researches which are similar to this work are discussed below.

i. Chen et al. [8] provided a top-of-the-line result on the Stanford Natural Language Inference Dataset using Long Short-Term Memory (LSTM). They employed Bi-directional LSTM (BiLSTM) as one of the building blocks. Later it is used to perform inference composition to construct the final prediction.

ii. Kim et al. [9] propose a densely-connected co-attentive recurrent neural network to find semantic relations between sentences. To overcome the problem of the extremely huge size of the feature vector due to densely connected networks, they also proposed an autoencoder after dense concatenation.

iii. Nguyen and Smeulders [10] incorporated clustering into active learning. Their method first constructs a classifier on the set of the cluster representatives, and then, with the help of a local noise model, it passes the classification decision to the other samples. The model allows selecting the most representative samples as well as avoids labelling samples in the same cluster. The paper focuses on discriminative models, including logistic regression and Support Vector Machines (SVM), which are less sensitive to training data and hence, good for active learning.

iv. Ru et al. [11] propose adversarial uncertainty sampling in discrete space (AUSDS) which retrieves informative unlabeled samples in an effective manner and is 10x faster when compared to a classic uncertainty sampling method for active learning.

v. Schumann and Rehbein [12] showed that it is possible to use Membership Query Synthesis [5] for generating AL queries for natural language processing, using Variational Autoencoders for query generation, and provides competitive performance to pool-based AL strategies which significantly reduces annotation time.

vi. Ren et al. [13] propose a novel fake news detection framework ”Adversarial Active Learning based Heterogeneous Graph Neural Network for Fake News Detection (AA-HGNN)”, which provides a hierarchical attention mechanism to perform node representation learning in the HIN. In this paper, the authors model the news content and related entities as a News-HIN. The AA-HGNN utilizes both structural information as well as News-HIN to identify fake news.

vii. Xiang et al. [14] propose a novel approach which exploits linguistic regularities in profane language via statistical topic modeling on a huge Twitter corpus, and detects offensive tweets using these automatically generated features. This approach works with various Machine Learning models such as J48 decision tree learning, SVM, logistic regression (LR), and random forest (RF).

viii. Watanabe et al. [15] propose a pragmatic approach to collecting hateful speech. This approach uses unigrams and patterns that are automatically collected from a training dataset. An accuracy of 87.4% was achieved on detecting whether a tweet is offensive or not (binary classification) and 78.4% accuracy when detecting a tweet is hateful, offensive, or clean (ternary classification).

ix. Palakodety et al. [2] propose a classifier which can classify comments supporting the Rohingyas. This is done by building a corpus from YouTube comments and applying multiple AL strategies based on nearest-neighbors in the comment-embedding space.

Chapter 3

Classification of Comments Supporting the Indian Farmers' Protest

3.1 Overview

This chapter provides the proposed work done so far in the classification of comments supporting the Indian farmers' protest through active learning and weak supervision.

3.2 Dataset

To construct the dataset, 412,445 comments were parsed from the comment section of YouTube using the YouTube API (Application Programming Interface) which came from 1077 unique videos. A corpus of 16,842 comments was made by using a language filter which keeps texts of purely English characters and a keyword-based filter which chooses comments that have those keywords or their different spelling variations. This removes most of the ambiguous and spam comments from the dataset. Some of the keywords are given in Table 3.1. These comments may be written in either pure English or in some other language written in Latin script, like Hinglish, which is Hindi written using English alphabets. After that, basic text preprocessing is done, like contraction expansion, punctuation, url, username, and emoji removal, converting to lowercase and lemmatization as shown in Figure 3.1. Stop words were not removed as FastText can handle them and some of them signified stances like ”against” and ”not”. To make a labelled dataset for machine learning, 4478 comments were hand labelled, out of which 1846 were labelled 0, 1408 were labelled 1, and 1224 were labelled 2, which signifies comments Against the Farm Bills, In Support of the Farm Bills, and Neutral, respectively. Out of these three labels, comments of either label 0 or 1 were considered, i.e., either against or in support of the bill. To make the labelling process more user-friendly and to make comments of uniform length, each comment was trimmed up to 300 characters.

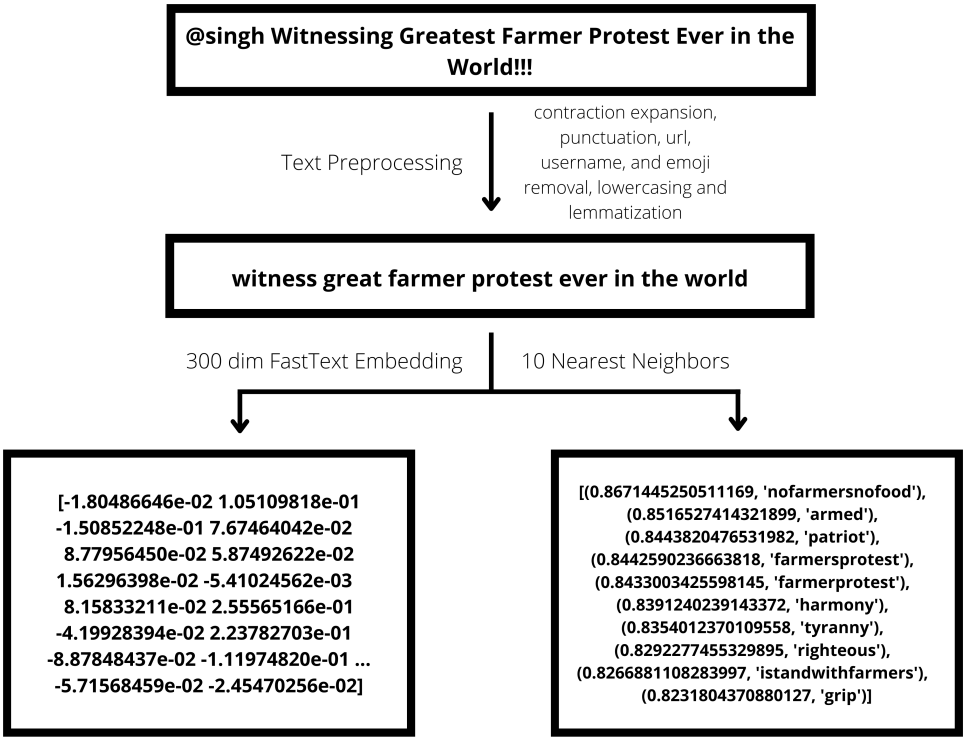


Figure 3.1: Text Preprocessing Flowchart

Table 3.1: Keyword Based Filtering List

Keywords
aatankwadi, against, apmc, bill bjp, bsp, congress, choukidar dacoit, dakaat, dakat, election farm, farmer, godi, haryana, india, kejriwal, khalistan, kisaan lakhimpur, majdoor, mandi, media modi, modiji, msp, pakistan price, punjab, rakesh, rally ravish, reject, repeal, rss rubbish, sapot, sarkar,support taliban, terrorist, tikait tractor, victory, yogi, yogiji zee, zindabad

3.3 Dataset Labelling

Dataset labelling is one of the most difficult and time-consuming tasks in the entire supervised machine learning process. AL solves this problem by automating the labelling process along with the user. Since the goal is to find the best results in classifying the comments, the entire dataset has to be manually labelled to compare the accuracy.

Labelling comments subjected to political debates is challenging as the person has to possess knowledge of current politics, laws, as well as grammar. Lack of these may result bias in the dataset. To minimize this effect, the labeling was done and cross checked by the authors of this paper. Table 3.2 shows some sample comments along with their true labels.

Table 3.2: Comments Along With Their Assigned Labels

No.	Comments	Labels
1.	Farmers protest is revolutionary and historical.	0
2.	Modhi don't like farmers. He want them be silent.	0
3.	Modiji affraid of UP Election	0
4.	The actual farmers are working in the field....and contributing to the nation	1
5.	I m support of bill	1
6.	Its not really anti farmers but anti middleman	1

3.4 Dataset Analysis

Dataset analysis is an important step before any kind of classification. Although the dataset is formed by comments from the people having a serious bone of contention, one thing is common: either they are in support of or against the bill. Figure 3.2 shows the wordcloud which shows the most frequent words in the dataset. The stance of a comment could be identified by analyzing the kinds of words present in it. Figure 3.3 shows the class-wise frequency of emotional words present in the dataset. Figures 3.4 and 3.5 show the frequency of frequent bigrams and trigrams, respectively.

Apart from emotional words, we can study the dataset by plotting the points (comments) with respect to the presence of a keyword or phrase in the comments. Figure 3.6 shows the plot of dataset comments based on TF-IDF scores of two phrases. Since the words may or may not be present in each comment, they are not a reliable source of classification criteria. However, they can be used for weak supervision which will be later discussed in the upcoming sections.

Figure 3.2: Wordcloud of the Dataset

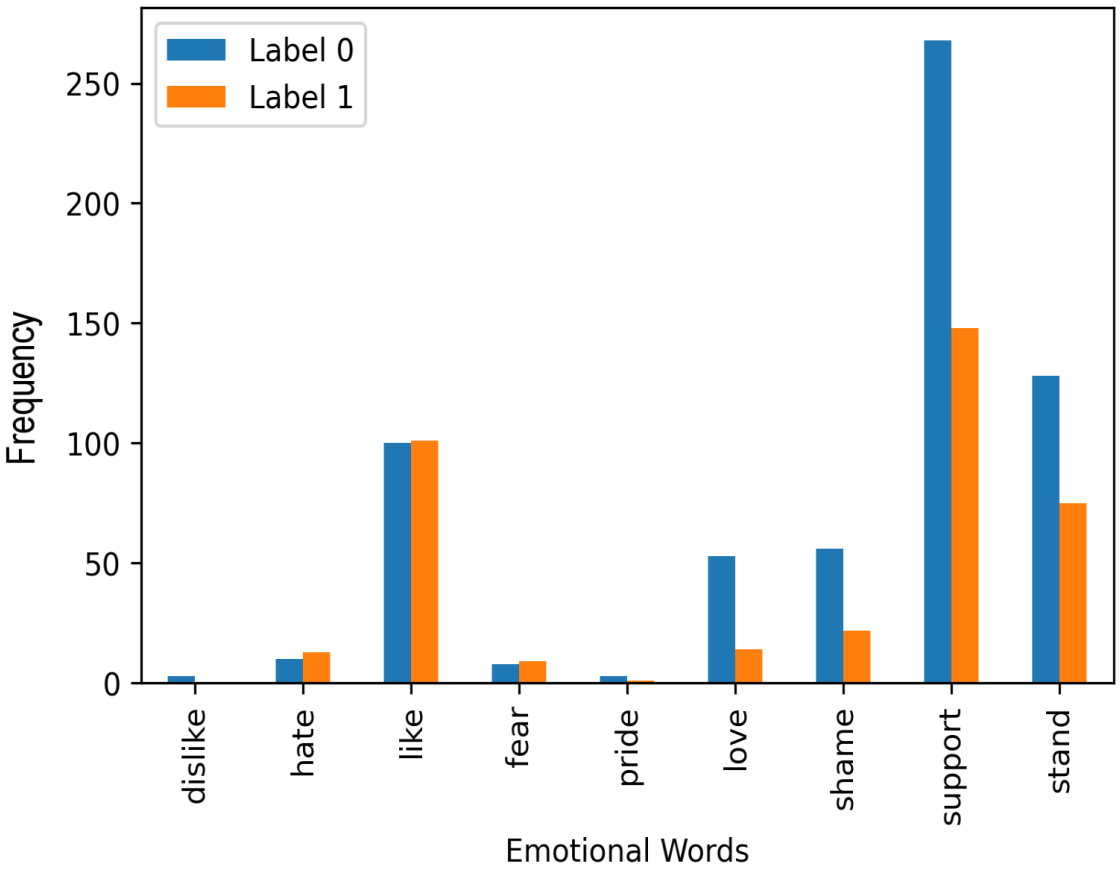


Figure 3.3: Frequency of Emotional Words in the Dataset

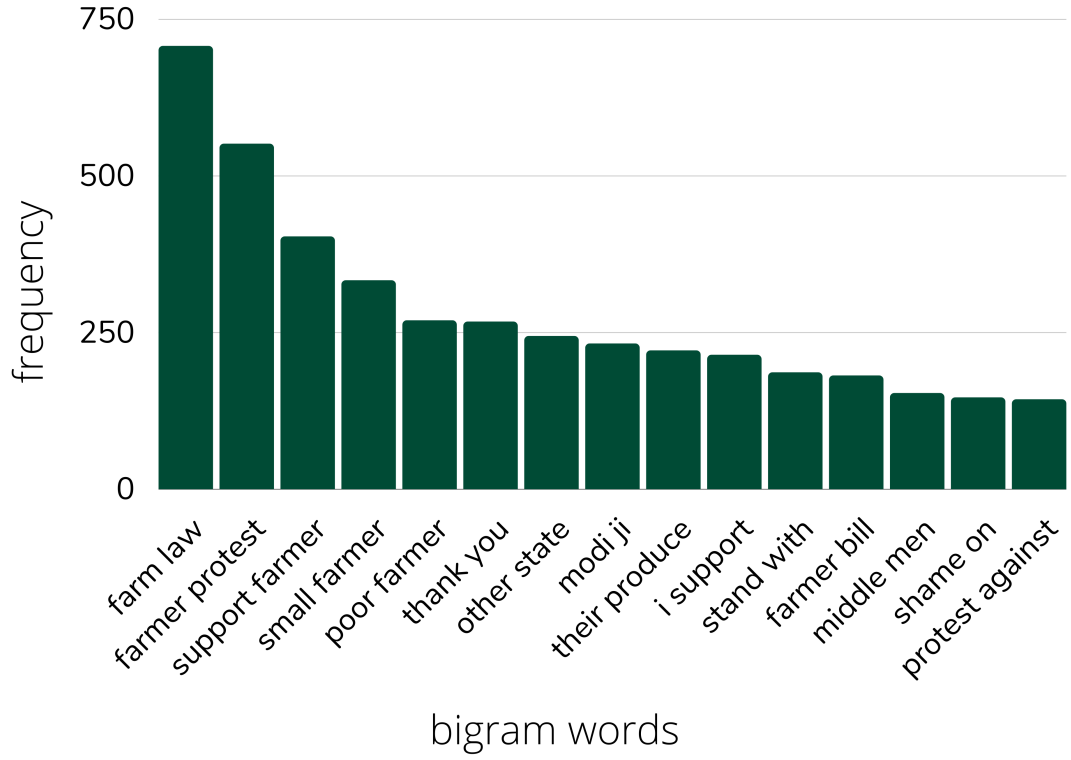


Figure 3.4: Frequency of Frequent Bigrams in the Dataset

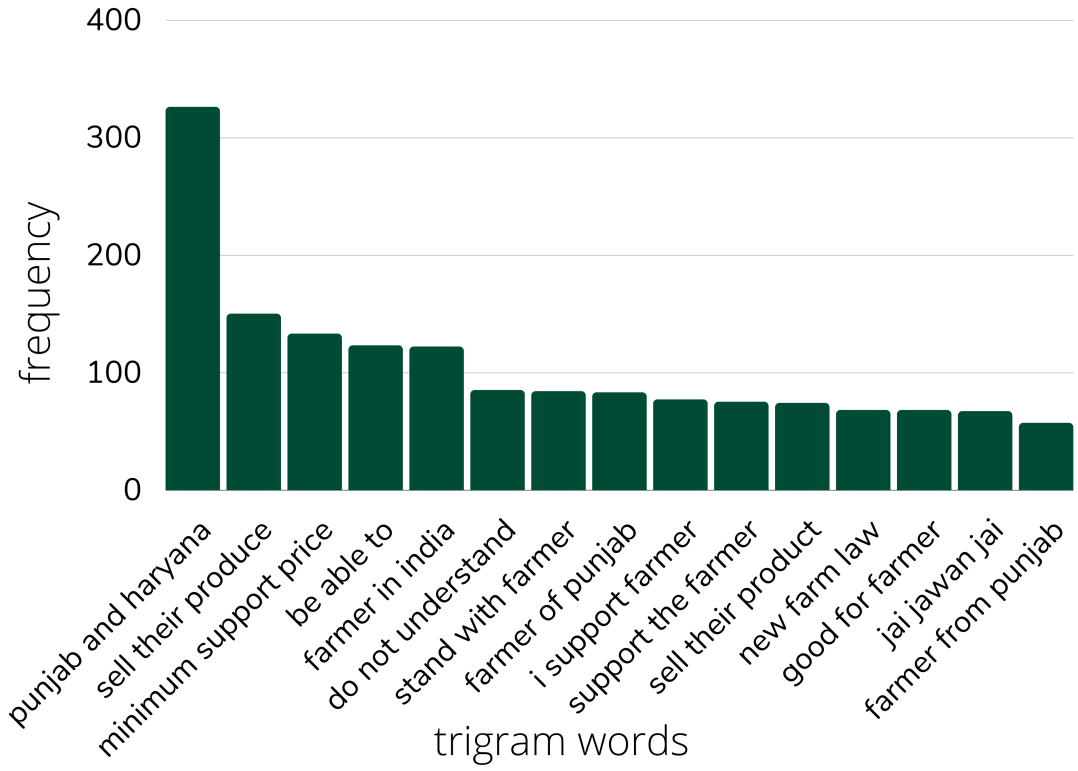
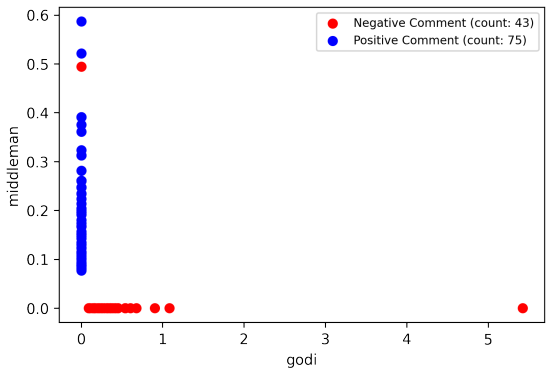
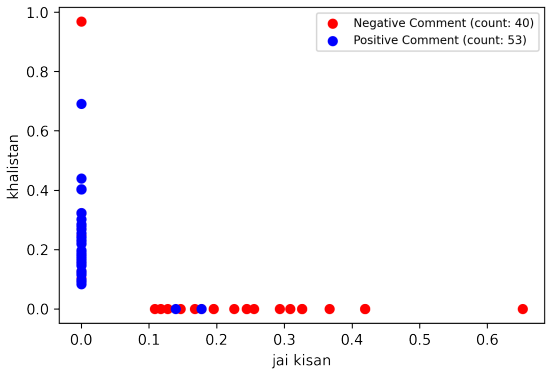


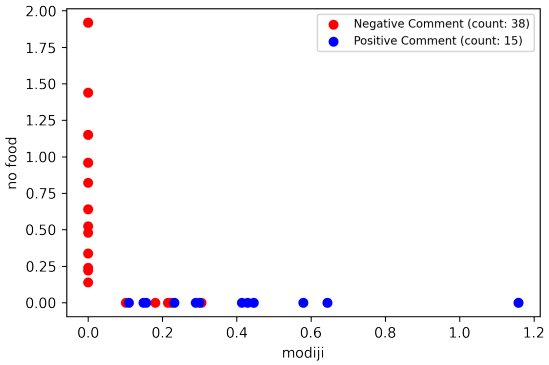
Figure 3.5: Frequency of Frequent Trigrams in the Dataset



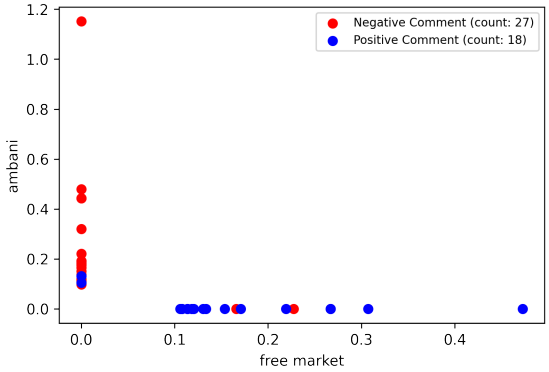
(a) Godi vs. Middleman



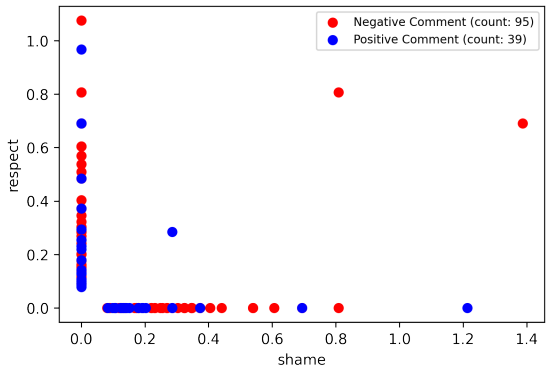
(b) Jai Kisan vs. Khalistan



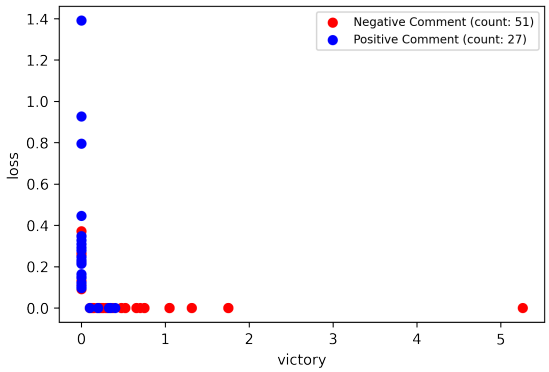
(c) Modiji vs. No Food



(d) Free Market vs. Ambani



(e) Shame vs. Respect



(f) Victory vs. Loss

Figure 3.6: Plotting Comments based on TF-IDF Scores Of Two Phrases

3.5 FastText Unsupervised Model

The corpus, which was created after text preprocessing, was fed to FastText [6] for unsupervised learning to generate two unsupervised models of Word N-Grams (N = 2 and N = 3). The model parameters were set to 300 dimensions, a 0.01 learning rate, and 50 epochs. All the analysis shown in the upcoming sections is done using the N=2 model.

3.6 Metrics and Classifiers

3.6.1 Performance Metrics

Four metrics were taken to evaluate the performance of the proposed method; accuracy, precision, f1 score, and recall, whose formulas are given in Equations 3.1, 3.2, 3.3, and 3.4 respectively. These are calculated with the help of the confusion matrix as shown in figure 3.7.

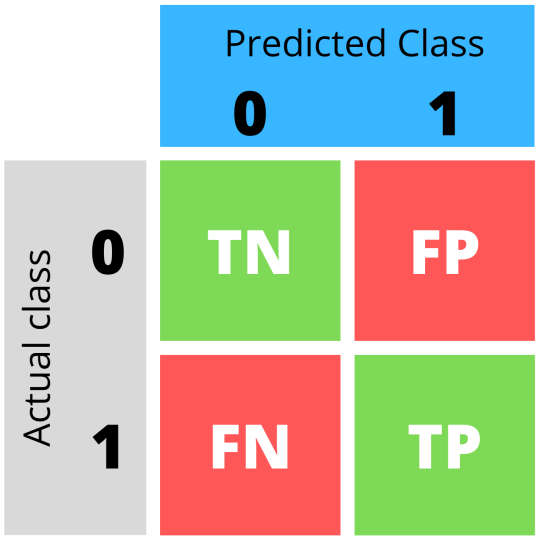


Figure 3.7: Confusion Matrix

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{3.3}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.4}$$

Here TP, TN, FP and FN means True Positives, True Negatives, False Positives and False Negatives respectively.

3.6.2 Hyperparameter Tuning and Verification

Validation curves are used to tune a single hyperparameter of a classifier, which can significantly increase the classifier's performance. It has two curves, an accuracy curve and the other a k-fold cross-validation curve. The plotting is performed by varying the value of the hyperparameter and plotting the training scores against the crossvalidation scores. A sweet spot for choosing an optimal value of hyperparameter will be the point where both curves have high accuracy as well as the gap between them is low. A low accuracy score denotes underfitting, whereas a high gap denotes overfitting.

3.6.3 Receiver Operating Characteristic (ROC) and Detection Error Tradeoff (DET) Curves

The ROC curve is the plot of True Positive Rate vs. False Positive Rate at different classification thresholds. The Area Under Curve (AUC) is the region below the curve that measures how well the predictions are ranked.

The DET curve is a plot of False Negative Rate vs. False Positive Rate. The DET curve allows easier analysis of classifiers because of the linear scale. The user can directly check at which False Positive Rate, the False Negative Rate is low and vice-versa.

3.7 Execution and Results

A split of 80-20 percent was used for training and testing, respectively. Out of 2603 training samples, 1477 are negatives (have label 0), i.e., comments that are against the bill, and 1126 are positives (have label 1). Then, out of 651 testing samples, 369 are negative and 282 are positive. After that, classification was performed using four classifiers, LR [16], SVM [17], K-Nearest Neighbors (KNN) [18], and Multi-Layer Perceptron (MLP) [19] by feeding them with the embeddings.

As AL reduces the effort of manual annotation, the authors decided to perform two AL techniques; Random Sampling and Uncertainty Sampling. First, the training data was randomly split into a seed set and an expansion set with a split of 1–99 percent respectively. The authors started from a tiny seed set and gradually expanded it with each iteration of batches. The seed set contains 15 negatives and 11 positive comments. The expansion set contains 2577 comments.

3.7.1 Random Sampling

In this sampling technique, batches of 20 comments were taken from the expanded set and comments were picked one by one. Then Nearest Neighbor (NN) vectors were formed each having 40 dimensions, using the fastText model, and cosine similarity was calculated for each comment against all the comments in the seed set. Whichever comment had the highest similarity score, the comment was classified into that class and the seed set was expanded using that comment. Thereafter, r number of comments were randomly assigned with their true labels by the oracle. Figure 3.8 shows the working of the random sampling technique and the results with $r = 4$ is shown in Table 3.3 along with the results of uncertainty sampling and weak classification, which will be discussed in the upcoming section.

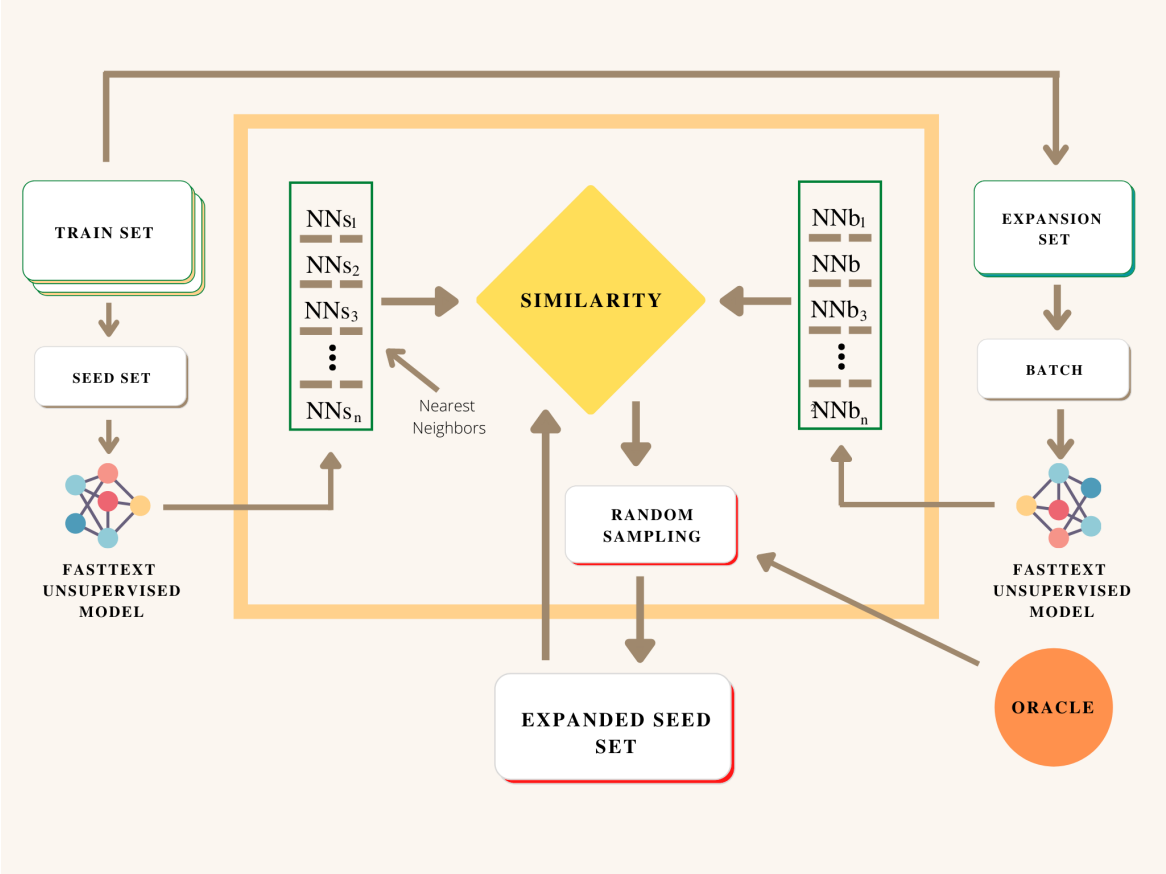


Figure 3.8: Batch-wise Nearest Neighbor Based Random Sampling

3.7.2 Uncertainty Sampling

In this sampling technique, just like random sampling, batches of 20 comments were taken, but instead of finding their nearest neighbors, first both the seed set and the expansion set were transformed using the unsupervised model, and then predictions were made on the transformed testing set using the seed set as training data. Sorting was done based on their

predicted probability score in non-decreasing order using the smallest-margin [20] technique as shown in equation 3.5. Here, the top u most uncertain comments are selected and labelled by the user. Rest of the comments are thrown away because they could be classified easily and won't help in classifying uncertain comments.

$$\phi_M(x) = P_\theta(y_0^*|x) - P_\theta(y_1^*|x) \tag{3.5}$$

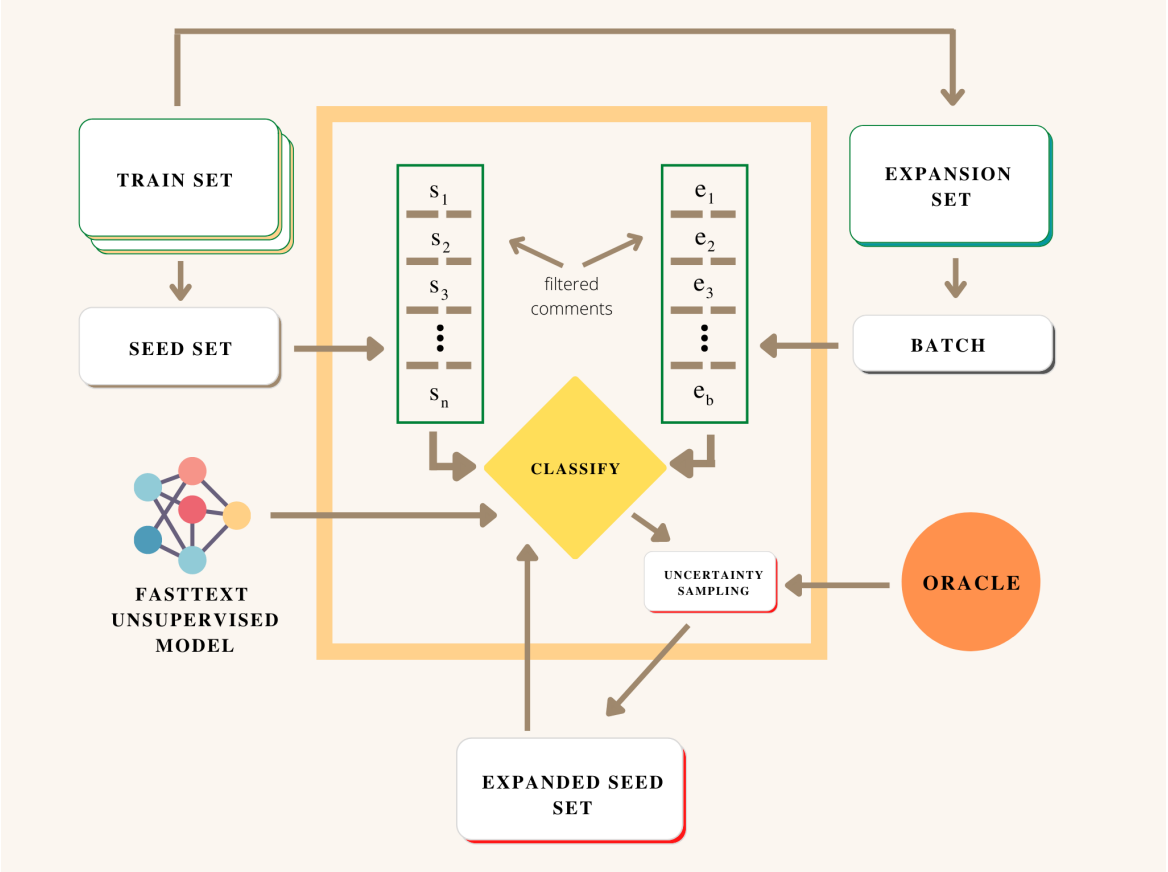
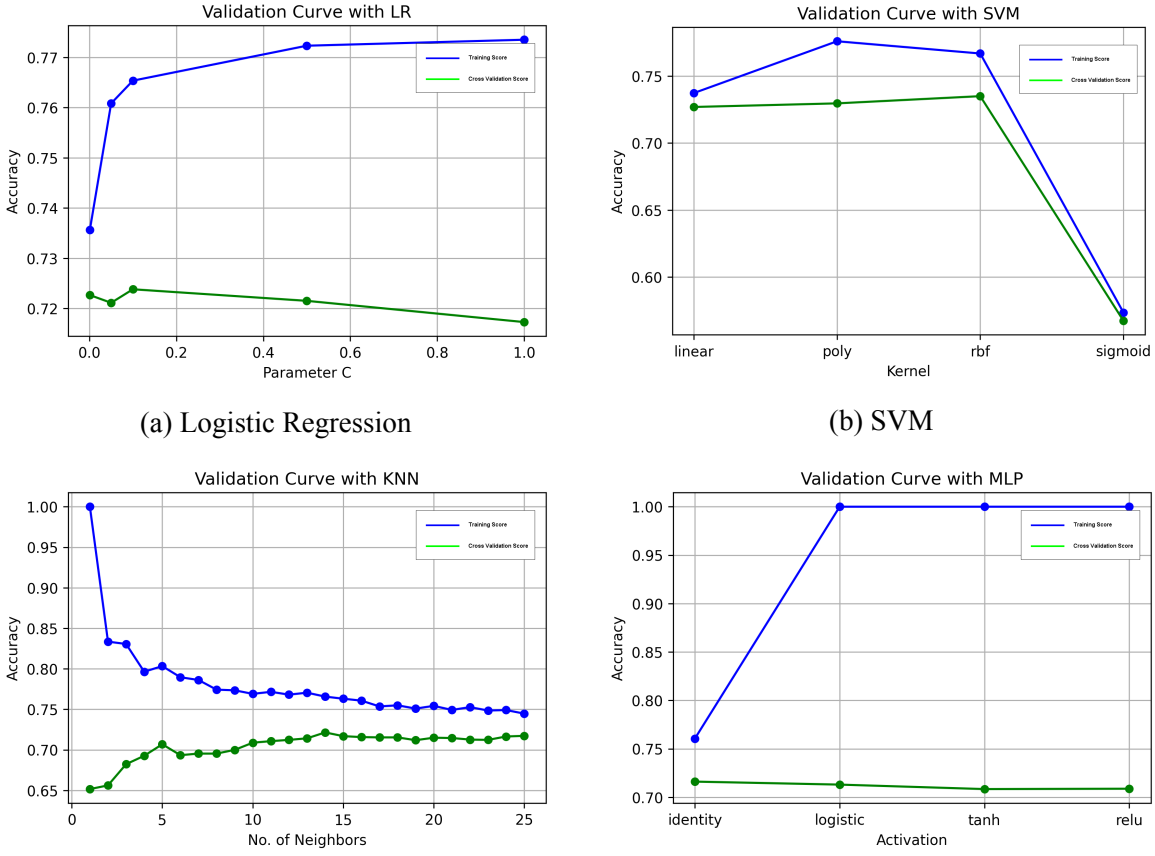


Figure 3.9: Batch-wise Word Embedding Based Uncertainty Sampling

The testing was performed by taking the top five most uncertain comments from each batch and appending them to the seed set. The batch size and the number of uncertain comments seems arbitrary, however, they were chosen after testing with different values of batch size and number of top uncertain comments, by taking accuracy, underfitting and overfitting into consideration. Figure 3.9 shows the batch-wise uncertainty sampling technique we used to expand the seed set. Figure 3.11 shows the validation curve of the four different classifiers. From validation curves, the hyper-parameters for the classifiers were set. For LR, SVM and MLP, first standard scaling [21] was performed. Hyper-parameters for LR were; solver: lbfgs [22], penalty: l2, C: 0.05, max iterations: 10000 and random states: 2. For SVM, the Radial Basis Function (RBF) Network [23] was chosen as the kernel. Like SVM, for KNN, only one hyperparameter was chosen, i.e., the number of neighbors k as 19,

and for MLP, the activation function was set to logistic with 1000 as the maximum number of iterations. Figure 3.12 shows the comparison of the ROC and DET curves of training data with the expanded set. Figure 3.13 and 3.14 show the ROC and DET curves of four different expanded seed set sizes formed by taking u as 1, 2, 5 and 10 respectively when classified using Logistic Regression, SVM, KNN, and MLP respectively. It was found that of the 666 comments, which were expanded using uncertainty sampling, which is 75% fewer comments than training data, a decent classification accuracy was seen. Figure 3.15 shows the accuracy, precision (weighted), recall (weighted), and f1 score (weighted) of classifying the test set at each step of expansion. Here the batch size was 20, and top five uncertain comments were picked and labelled. The confusion matrices of the proposed method are shown in Figure 3.16.



(a) Logistic Regression

(b) SVM

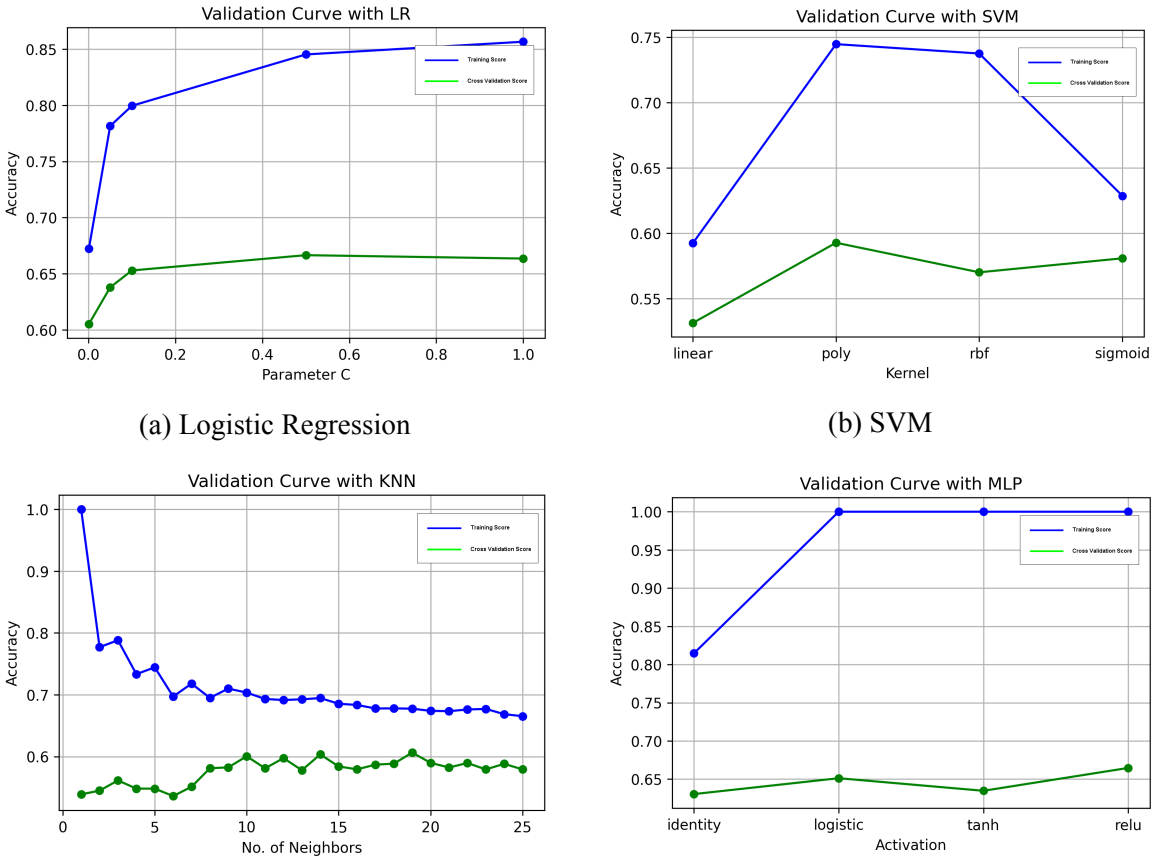
(c) KNN

(d) MLP

Figure 3.10: Validation Curves of Four Classifiers on Training Data

3.7.3 Weak Classification

Word embeddings makes text classification possible. However, sometimes some classifiers can't take full advantage of it. On analysing the dataset, it was found out that many comments from both sides had some common phrases present in them. The presence of these phrases in comments could act as features and weak classification could be performed. Figure 3.6



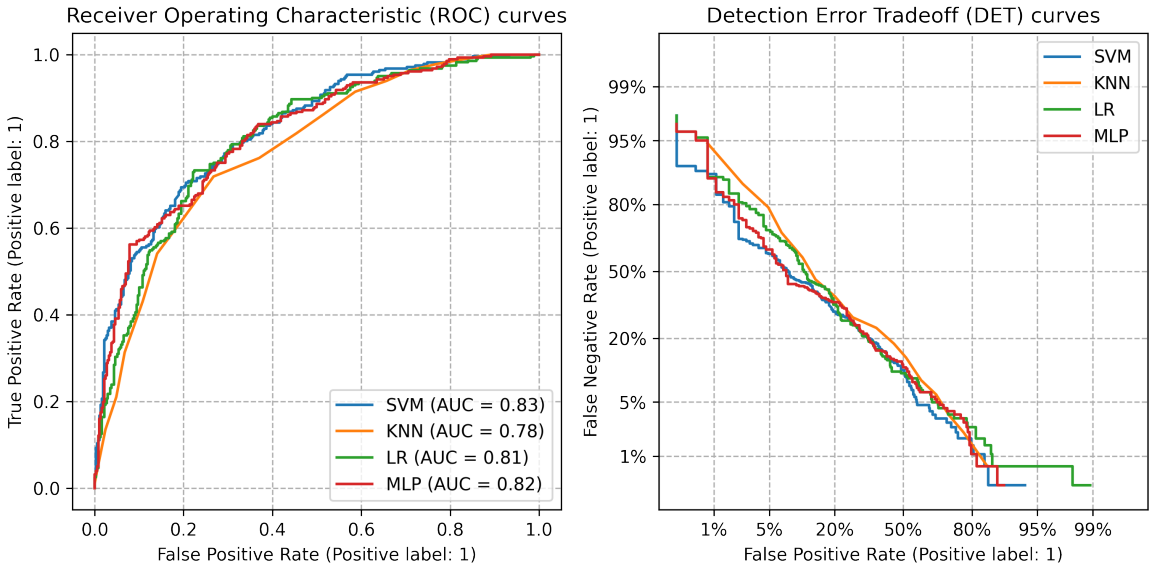
(a) Logistic Regression

(b) SVM

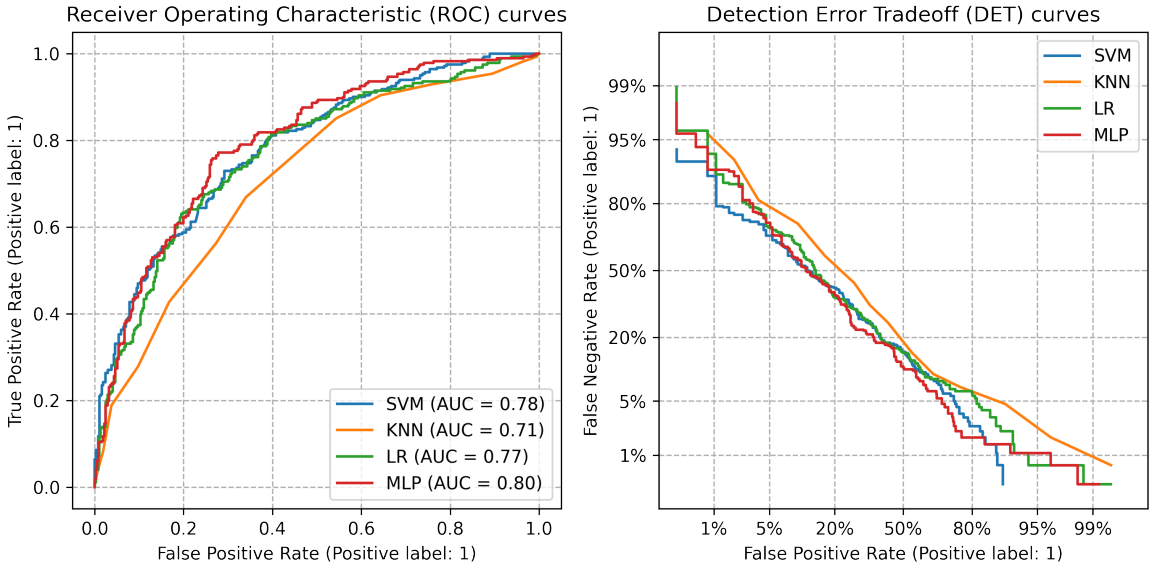
(c) KNN

(d) MLP

Figure 3.11: Validation Curves of Four Classifiers on Expanded Set

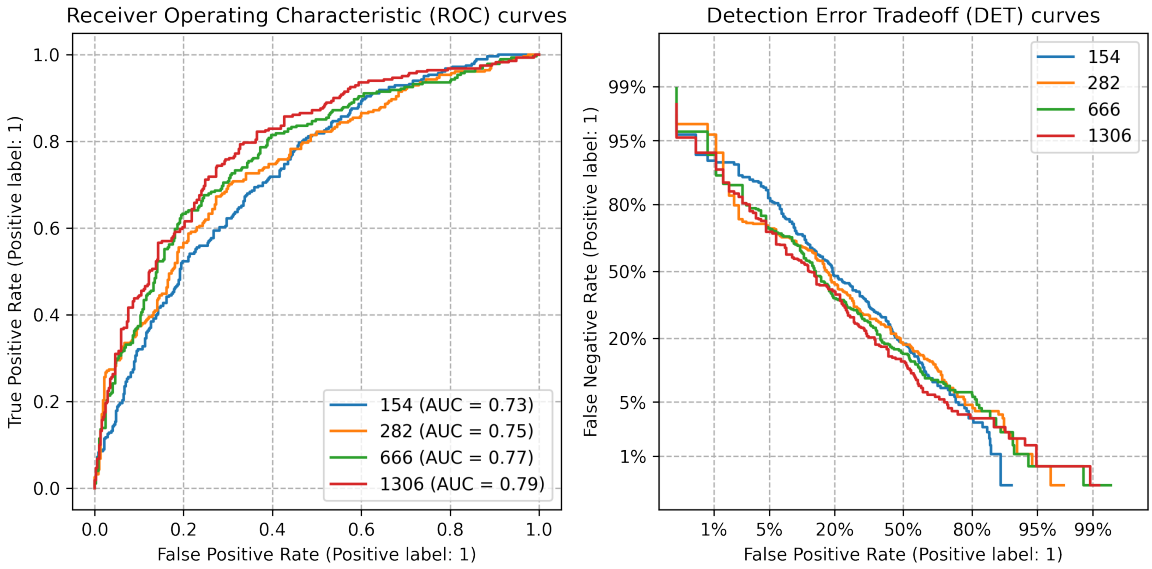


(a) Training Data

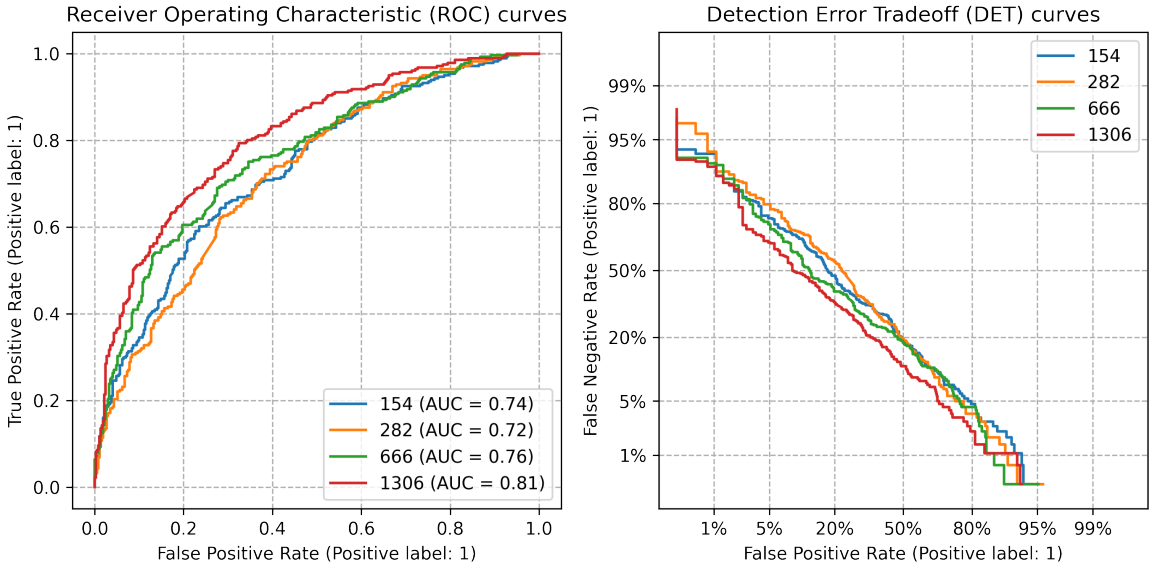


(b) Uncertainty Sampled Data

Figure 3.12: Comparative ROC and DET curves of Four Classifiers on Training and Uncertainty Sampled Data)

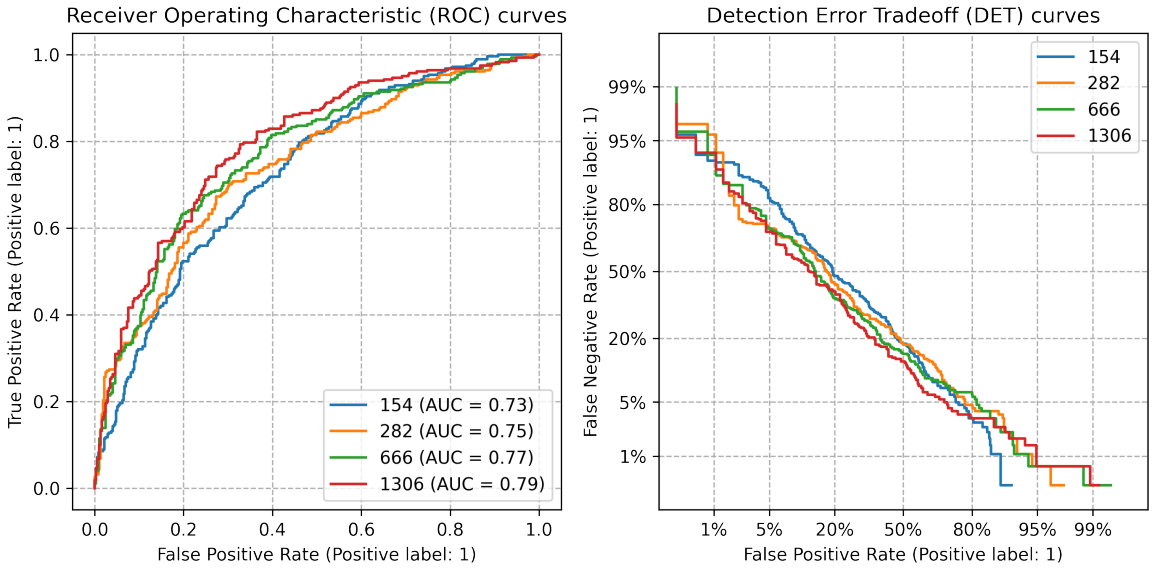


(a) Logistic Regression

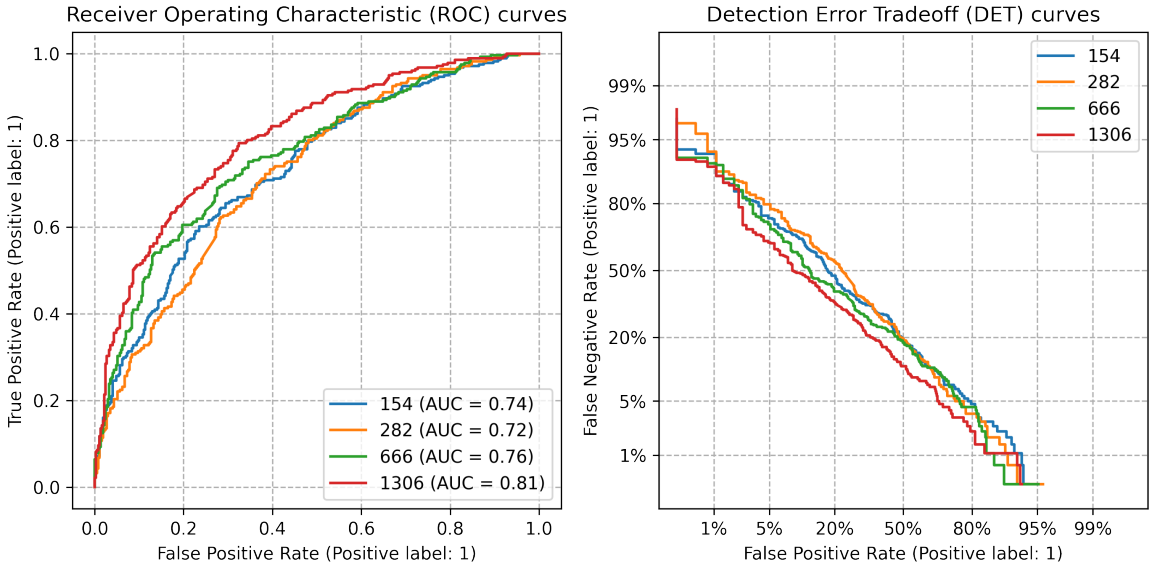


(b) SVM

Figure 3.13: Comparative ROC and DET curves of Logistic Regression and SVM on Different Sizes of Expanded Set

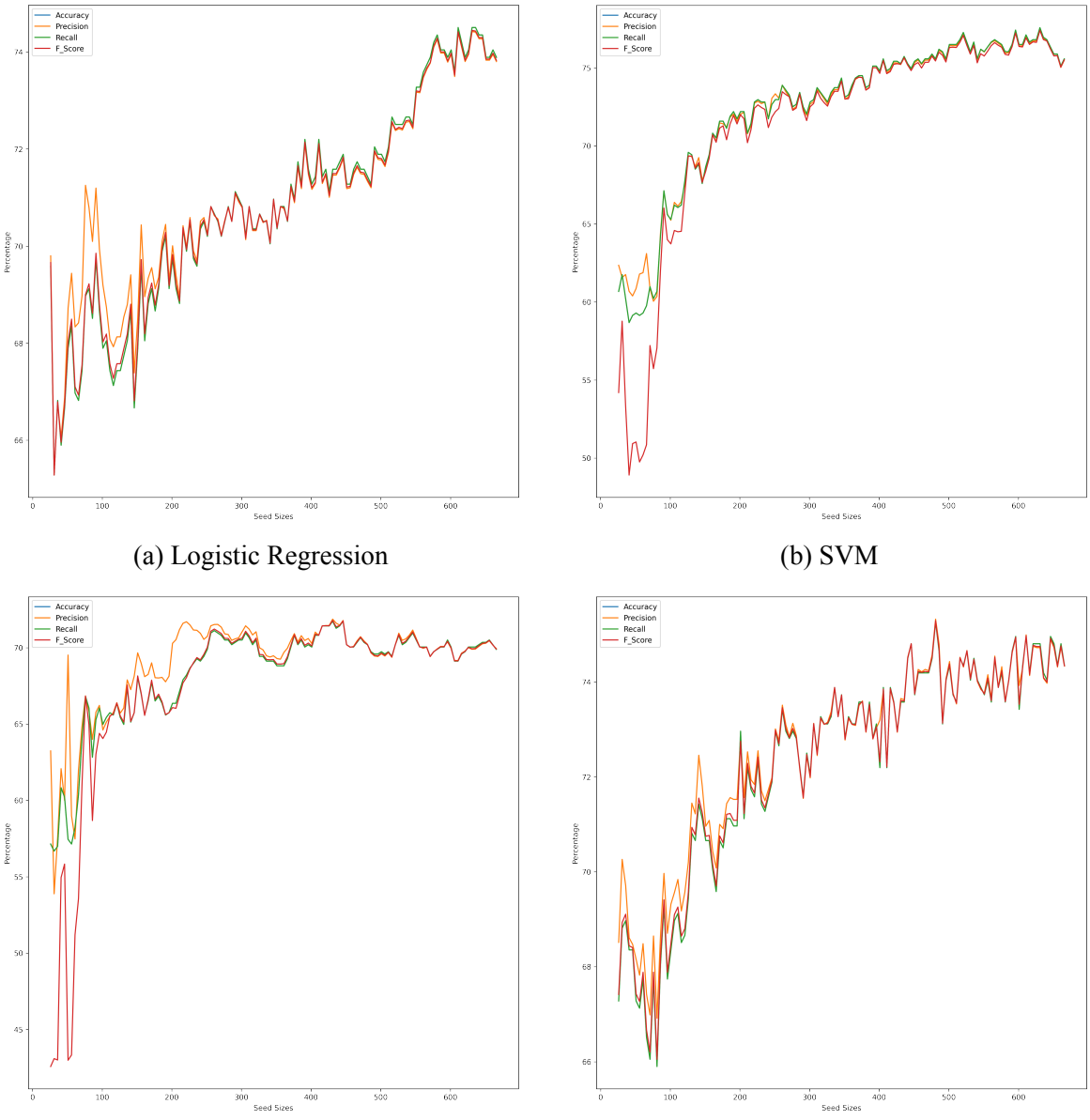


(a) KNN



(b) MLP

Figure 3.14: Comparative ROC and DET curves of KNN and MLP on Different Sizes of Expanded Set



(a) Logistic Regression

(b) SVM

(c) KNN

(d) MLP

Figure 3.15: Classification Results During Seed Set Expansion

		Supervised Learning			Random Sampling			Uncertainty Sampling	
		Predicted			Predicted			Predicted	
		0	1		0	1		0	1
LR	Actual 0	293	77	0	283	87	0	301	69
	1	85	196	1	102	179	1	65	216
SVM	Actual 0	312	58	0	293	77	0	326	44
	1	79	202	1	114	167	1	69	212
KNN	Actual 0	251	119	0	236	134	0	239	131
	1	69	212	1	80	201	1	65	216
MLP	Actual 0	296	74	0	276	94	0	313	57
	1	76	205	1	102	179	1	59	222

Figure 3.16: Comparison of Confusion Matrices for Three Different Techniques

shows that positive and negative comments could be distinguished based on TF-IDF scores of the two phrases. So, in order to do weak classification, FastText word embeddings were combined with the weak labels as shown in Figure 3.17, and uncertainty sampling was performed. After the incorporation of weak classification, the results of KNN classifier have been improved from 69.89% to 72.66% for WordNGrams=2, as shown in Table 3.3.

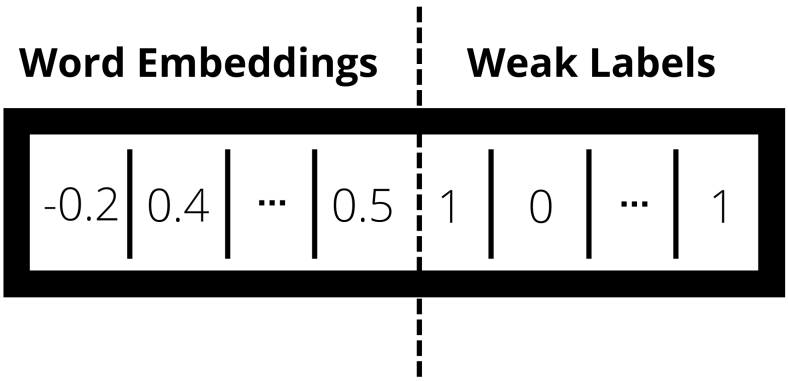


Figure 3.17: Weak Classification Feature Vector

3.7.4 Comparison With Existing Work

Table 3.4 summarises the results of Palakodety et al.'s [2] work. They first began with a small seed set and then expanded it using random sampling, certainty sampling, and uncertainty sampling. Then they used SVM (n-gram + embeddings) to further improve their results. In the proposed work, the authors managed to get almost similar, or sometimes even better, results when compared to classification without using AL.

Table 3.3: Classification Results of the Proposed Method

Algorithm	**WordNGrams**	**Accuracy %**	**Precision %**	**F1 score %**	**Recall %**
LR	N=2	75.12	71.80	69.75	70.76
SVM		78.96	77.69	71.88	74.68
KNN		71.12	64.04	75.44	69.29
MLP		76.96	73.48	72.95	73.21
LR	N=3	75.88	72.97	70.11	71.51
SVM		78.80	76.38	73.67	75.00
KNN		71.12	63.80	76.51	69.58
MLP		77.11	73.57	73.31	73.44
LR + Random + NN	N = 2	70.97	67.30	63.70	65.45
SVM + Random + NN		70.66	68.44	59.43	63.62
KNN + Random + NN		67.13	60.00	71.53	65.26
MLP + Random + NN		69.89	65.57	63.70	64.62
LR + Random + NN	N = 3	68.97	64.91	61.21	63.00
SVM + Random + NN		70.35	67.46	60.50	63.79
KNN + Random + NN		66.66	59.82	69.40	64.25
MLP + Random + NN		69.65	62.04	66.41	64.15
LR + Uncertainty	N = 2	79.42	75.79	76.87	76.33
SVM + Uncertainty		82.64	82.81	75.44	78.96
KNN + Uncertainty		69.89	62.25	76.87	68.79
MLP + Uncertainty		82.18	79.57	79.04	79.29
LR + Uncertainty	N = 3	77.58	74.19	73.67	73.93
SVM + Uncertainty		82.49	79.93	79.36	79.64
KNN + Uncertainty		69.74	62.88	72.95	67.54
MLP + Uncertainty		81.72	77.94	77.06	77.50
LR + Uncertainty + Weak	N = 2	74.81	70.10	72.60	71.33
SVM + Uncertainty + Weak		78.50	74.74	75.80	75.26
KNN + Uncertainty + Weak		72.66	66.04	75.44	70.43
MLP + Uncertainty + Weak		74.81	70.24	72.24	71.23
LR + Uncertainty + Weak	N = 3	73.57	68.85	70.82	69.82
SVM + Uncertainty + Weak		77.42	73.60	74.37	73.98
KNN + Uncertainty + Weak		68.35	62.80	65.48	64.11
MLP + Uncertainty + Weak		76.19	73.51	70.11	71.77

Table 3.4: Voice-for-the-voiceless classifier performance

Performance Measure	Seed set + random sampling + NN in the embedding space	Uncertainty sampling	SVM (n grams + embeddings)
Precision	67.17 ± 9.90%	73.65 ± 3.45%	76.49 ± 3.41%
Recall	32.35 ± 7.65%	79.39 ± 3.72%	80.30 ± 3.73%
Accuracy	82.04 ± 2.34%	75.38 ± 2.76%	77.71 ± 2.56%
F1 score	43.02 ± 7.90%	76.34 ± 2.77%	78.28 ± 2.71%
AUC	83.61 ± 2.88%	83.67 ± 2.61%	85.91 ± 2.32 %

Chapter 4

Conclusion

In this thesis, we provided active learning-based approaches to classify comments that support the farmers' protest with those that are against the farmers' protest. We built our own dataset by parsing YouTube comments, and that provided results close to real-life scenarios. The batch-wise expansion technique allowed us to expand seed set comments in a controlled fashion without degrading the classifier's performance. The validation curves helped in tuning the hyper-parameters and the ROC and DET curves provided the performance of each classifier, among which SVM performed the best. The final expanded seed set only had 25% of the original training data that was hand labelled but still had enough to provide decent results, sometimes even better when compared to training data when all labels were present. From the results, we can conclude that we have successfully classified comments using active learning-based techniques with decent accuracy, and the methods we showed here can be extended to several other domains. When compared to the Voice-for-the-voiceless classifier, we discovered that our method and dataset produce comparable, if not better, results, which further validates our work. This work also enhanced our socio-political knowledge and we hope that it will also enlighten all people around the globe about this affair.

Scope for Further Research

The proposed work was restricted to YouTube only. In the future, we will try to fetch comments from various other sources as well as check their results with more classifiers.

References

[1] Shankar, A., 2021. "Indian agriculture farm acts: 2020". *International Journal of Modern Agriculture,* **10**(2), pp. 2907–2914.

[2] Palakodety, S., KhudaBukhsh, A. R., and Carbonell, J. G., 2020. "Voice for the voiceless: Active sampling to detect comments supporting the rohingyas". In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, pp. 454–462.

[3] Lewis, D. D., and Gale, W. A., 1994. "A sequential algorithm for training text classifiers". In SIGIR'94, Springer, pp. 3–12.

[4] Zhu, X. J., 2005. "Semi-supervised learning literature survey".

[5] Settles, B., Craven, M., and Friedland, L., 2008. "Active learning with real annotation costs". In Proceedings of the NIPS workshop on cost-sensitive learning, Vol. 1, Vancouver, CA:.

[6] Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T., 2016. "Fasttext. zip: Compressing text classification models". *arXiv preprint arXiv:1612.03651*.

[7] Mikolov, T., Chen, K., Corrado, G., and Dean, J., 2013. "Efficient estimation of word representations in vector space". *arXiv preprint arXiv:1301.3781*.

[8] Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., and Inkpen, D., 2016. "Enhanced lstm for natural language inference". *arXiv preprint arXiv:1609.06038*.

[9] Kim, S., Kang, I., and Kwak, N., 2019. "Semantic sentence matching with densely-connected recurrent and co-attentive information". In Proceedings of the AAAI conference on artificial intelligence, Vol. 33, pp. 6586–6593.

[10] Nguyen, H. T., and Smeulders, A., 2004. "Active learning using pre-clustering". In Proceedings of the twenty-first international conference on Machine learning, p. 79.

[11] Ru, D., Feng, J., Qiu, L., Zhou, H., Wang, M., Zhang, W., Yu, Y., and Li, L., 2020. "Active sentence learning by adversarial uncertainty sampling in discrete space". *arXiv preprint arXiv:2004.08046*.

[12] Schumann, R., and Rehbein, I., 2019. "Active learning via membership query synthesis for semi-supervised sentence classification". In Proceedings of the 23rd conference on computational natural language learning (CoNLL), pp. 472–481.

[13] Ren, Y., Wang, B., Zhang, J., and Chang, Y., 2020. "Adversarial active learning based heterogeneous graph neural network for fake news detection". In 2020 IEEE International Conference on Data Mining (ICDM), IEEE, pp. 452–461.

[14] Xiang, G., Fan, B., Wang, L., Hong, J., and Rose, C., 2012. "Detecting offensive tweets via topical feature discovery over a large scale twitter corpus". In Proceedings of the 21st ACM international conference on Information and knowledge management, pp. 1980–1984.

[15] Watanabe, H., Bouazizi, M., and Ohtsuki, T., 2018. "Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection". *IEEE access,* **6**, pp. 13825–13835.

[16] Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., and Klein, M., 2002. *Logistic regression*. Springer.

[17] Cortes, C., and Vapnik, V., 1995. "Support vector machine". *Machine learning,* **20**(3), pp. 273–297.

[18] Zhang, M.-L., and Zhou, Z.-H., 2005. "A k-nearest neighbor based algorithm for multi-label classification". In 2005 IEEE international conference on granular computing, Vol. 2, IEEE, pp. 718–721.

[19] Baum, E. B., 1988. "On the capabilities of multilayer perceptrons". *Journal of complexity,* **4**(3), pp. 193–215.

[20] Scheffer, T., Decomain, C., and Wrobel, S., 2001. "Active hidden markov models for information extraction". In International Symposium on Intelligent Data Analysis, Springer, pp. 309–318.

[21] Ahsan, M. M., Mahmud, M., Saha, P. K., Gupta, K. D., and Siddique, Z., 2021. "Effect of data scaling methods on machine learning algorithms and model performance". *Technologies,* **9**(3), p. 52.

[22] Saputro, D. R. S., and Widyaningsih, P., 2017. "Limited memory broyden-fletcher-goldfarb-shanno (l-bfgs) method for the parameter estimation on geographically weighted ordinal logistic regression model (gwolr)". In AIP Conference Proceedings, Vol. 1868, AIP Publishing LLC, p. 040009.

[23] Park, J., and Sandberg, I. W., 1991. "Universal approximation using radial-basis-function networks". *Neural computation,* **3**(2), pp. 246–257.