SN Computer Science (2022) 3:49
https://doi.org/10.1007/s42979-021-00922-z

ORIGINAL RESEARCH

Classification of Textual Sentiment Using Ensemble Technique

Md. Mashiur Rahaman Mamun[1] · Omar Sharif[1] · Mohammed Moshiul Hoque[1]

Received: 30 July 2021 / Accepted: 3 October 2021 / Published online: 5 November 2021

Abstract
In recent years, the widespread use of the Internet has resulted in a revolutionary way for people to share their feelings or sentiment on blogs, social media, e-commerce sites, and online platforms. Most of the feelings expressed on the online platforms are in textual forms (such as status, tweets, comments, and reviews). These textual expressions are unstructured, laborious, and time-consuming to organize, manipulate, or efficient storage due to their messy forms. Textual sentiment analysis refers to the automatic process of assigning an expression or text to an appropriate polarity (positive, negative, and neutral). Although Bengali is ranked seventh most popular language globally and the second famous Indic language, the development of language processing tools is minimal to date. This paper proposes an ensemble-based technique to classify Bengali textual sentiment into two categories: positive and negative. Due to the unavailability of the Bengali sentiment corpus, this work also developed a dataset (called 'Bengali Sentiment Analysis Dataset or BSaD') containing 8122 text expressions. This work investigates eight popular baseline classifiers [such as Logistic Regression (LR), Randon Forest (RF), Decision Tree (DT), K-nearest Neighbor (KNN), Support Vector Machine (SVM), Multinomial Naive Bayes (MNB), Stochastic Gradient Descent, and AdaBoost] with Term frequency-Inverse document frequency (TF-IDF) and Bag-of-words (BoW) feature for textual sentiment analysis on three datasets. This work also investigates the four ensemble methods (LR + RF, RF + SVM, LR + SVM, and LR + RF + SVM) developed by combining three best-performing base classifiers (LR, RF, and SVM). Experimental results show that the ensemble approach (i.e., LR + RF + SVM) with TF-IDF (uni-gram + bi-gram + tri-gram) features outperformed the other classifier models achieving the highest accuracy 82% on the developed dataset.

Keywords Natural language processing · Textual sentiment analysis · Feature extraction · Machine learning · Ensemble

Introduction

Sentiment analysis or classification is an automatic process that strives to uncover a user's viewpoint towards a particular entity. It intends to ascertain the contextual polarity of the textual contents (such as comments, posts, or opinions) as the neutral, negative, and positive [4, 13]. The proliferation of Internet usage through various social media platforms, micro-blogs, news portals, and e-commerce sites has resulted in enormous textual interactions. In these textual interactions, people express their feelings, emotions, opinions, and feedback via textual comments, tweets, reviews, posts, and concerns. Thus, analyzing this growing amount of textual expressions has gained much attention from several organizations due to its various practical applications. However, most textual expressions are unstructured, arduous, and time-consuming to manipulate, sort, and organize due to their messy form. Due to the fast and cost-effective nature, automatic sentiment classification has gained enhanced attention from several organizations. Recently, many organizations use sentiment classifiers for a broad range of purposes, like product analytics, brand monitoring, business research, customer service, social media analysis, and many more [40].

This article is part of the topical collection "Enabling Innovative Computational Intelligence Technologies for IOT" guest edited by Omer Rana, Rajiv Misra, Alexander Pfeiffer, Luigi Troiano, and Nishtha Kesswani.

✉ Mohammed Moshiul Hoque
moshiul_240@cuet.ac.bd

Md. Mashiur Rahaman Mamun
mamun.cse.71@gmail.com

Omar Sharif
omar.sharif@cuet.ac.bd

[1] Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong 4349, Bangladesh

Recently, it has been observed that the unstructured textual expressions, including comments, opinions, and reviews, have increased dramatically in the Bengali language. Although several studies have been conducted on high-resource languages, the sentiment analysis on the Bengali language is still in its infancy. The inadequacy of sentiment data, linguistics resources, and practical natural language processing (NLP) tools are critical hurdles to developing a Bengali sentiment analysis system. Machine learning (ML) techniques are used extensively in solving text classification problems, including sentiment analysis, due to their more accurate results than the human-crafted expert systems. A few studies attempted to analyze sentiment in Bengali using ML techniques (such as SVM, NB, DT, and RF). Most of these studies developed a classifier model on a small dataset and classification task performed on a specific domain such as book reviews [13], restaurant reviews [15, 30], and social media status [16]. Thus, the previous systems suffer from lower accuracy and generability. This work proposes an ML-based ensemble technique on a larger dataset to classify sentiment into positive and negative classes concerning textual contents to address the current constraints on sentiment classification in Bengali. Ensemble technique has already proven to be successful in a wide range of problem areas and real-world applications [1]. The reason behind this success is its ability to reduce variance between base classifiers in several scenarios, including feature selection, confidence estimation, error correction, and class imbalance problem [2]. Owing to these reasons, this work employed an ensemble technique. This technique consists of applying diverse classifiers and consolidating their predictions to train a meta-learning model. The ensemble is typically adopted to improve the performance of a particular method [10]. The specific contribution of this work is as follows:

- Develop a sentiment classification dataset for analyzing sentiment (called *BSaD*) containing 8122 Bengali texts.
- Propose an ensemble-based technique (using LR, RF, and SVM) to analyze textual sentiment in Bengali into two sentiment polarities: positive and negative.
- Investigate the sentiment analysis task performance using various machine learning techniques (LR, DT, RF, KNN, MNB, SVM, SGD, and AdaBoost) and ensemble methods (LR + RF, RF + SVM, LR + SVM, and LR + RF + SVM) on three datasets and employing the comparative analysis between the proposed model and existing techniques of Bengali textual sentiment analysis.

Related Work

Analyzing textual sentiment is a widely studied research concern in high-resource languages. Lai et al. [17] proposed a fine-grained emotion classification using graph convolution networks. Their method was applied for the emotion classification of Chinese micro-blogs and achieved 82.32% of micro-F-score. However, it remained unexplored how the method works for other languages, including low-resources. Luo [19] proposed a GRU (gated recurrent unit)-CNN (convolutional neural network)-based method with Latent Dirichlet Allocation (LDA) representation for analysis sentiment of online media texts. This work was evaluated in small datasets, and an in-depth analysis of the proposed model's performance is absent. Prabowo and Thelwall [22] proposed a hybrid method for sentiment classification from movie and product reviews. This method requires thousands of rules for classification, which is very difficult to generate and maintain. Mamta et al. [21] developed an ensemble technique using CNN, LSTM (long short-term memory), and GRU for sentiment analysis. Their method achieved 84.65% accuracy on Tweeter texts. Gamal et al. [11] investigated ten ML algorithms for sentiment classification tasks on IMDB, Cornell Movies, Amazon, and Twitter datasets. Their analysis revealed that the passive-aggressive technique achieved the highest accuracy (up to 96.96%) for all datasets. Amrani et al. [3] used bag-of-words (BOW) features to perform sentiment analysis on an Amazon product review dataset. A combined approach (RF + SVM) achieved the highest accuracy of 83.4%. A graph-based technique is used to classify opinion, which achieved an accuracy of 85.78% on Hindi EmotionNet [12]. The system cannot classify implicit emotions and cannot handle the text's contextual and semantic features.

Bengali is considered the seventh most widely spoken language globally. Nevertheless, the research on Bengali text processing is still in their infancy, especially in textual sentiment classification due to the unavailability of necessary resources and language processing tools [9]. Various ML techniques have been utilized for textual sentiment classification in Bengali, such as Multi-nomial naïve bayes (MNB) [16], SVM [35], and RF [33]. These works were evaluated on a limited dataset with positive and negative classes. Thus, their effectiveness and generability are unexplored. Hossain et al. [13] proposed a sentiment analysis model using MNB with uni-gram features. This work achieved the highest accuracy of 84% on 2000 book reviews. Chowdhury et al. [6] presented a sentiment analysis model for Bengali movie reviews and achieved an accuracy of 88.90% (for SVM) and 82.42% [for long short-term memory (LSTM)] on 4000 Bengali reviews. Sarkar [25] proposed an LSTM-based sentiment analysis to classify Bengali 1500 tweets into positive, negative, and neutral classes with an accuracy of 55.23%. Wahid et al. [38] proposed a sentiment analysis model using LSTM to classify the Bengali text into positive, negative, and neutral classes with an accuracy of 95% over a dataset consists of 10,000 Facebook comments. Sharif et al. [30] performed sentiment analysis on restaurant reviews which achieved an accuracy of 80.48% on 1000 reviews. Sarkar and Bhowmick [27] performed the sentiment analysis on the Bengali tweet dataset, where SVM and MNB classifiers were used for the classification. This work used *n*-gram and SentiWordnet features and gained a lower accuracy (45%) on a dataset containing approximately 1000 labelled tweets.

Investigating past studies in Bengali revealed that most sentiment analyses performed on a small dataset with a particular technique and considered only a specific domain. Thus, it remained unexplored how an ML model can develop to classify textual sentiment concerning multiple domains. Moreover, none of the previous work employed ML-based ensemble techniques for textual sentiment analysis in Bengali. By considering the constraints of past studies, this work proposes an ensemble-based approach for textual sentiment classification into positive and negative polarity. The proposed method is evaluated on three different datasets with more data than past studies.

BSaD: Bengali Sentiment Analysis Dataset

Due to the unavailability of the benchmark textual sentiment datasets in Bengali, this work developed a dataset (i.e., BSaD) to perform the sentiment analysis task. We followed the directions suggested by Das et al. [8] for developing the dataset. Following steps are carried out to prepare the BSaD:

- **Data Accumulation and Preprocessing**: The textual sentiment texts are collected from online news portals, Facebook posts/comments, Youtube comments, and blogs. Five human crawlers accumulated 8815 text documents over the approximately 16 month period (from August 2019 to December 2020). An automatic filter cleans the collected raw texts to reduce the annotation complexity and inequalities. Texts containing non-Bengali words and duplicate data have been removed. Moreover, texts with a length of smaller than two words are also discarded. A total of 8535 text documents are included in the dataset after completing the preprocessing and send for the human annotation.
- **Data Annotation**: Five postgraduate NLP enthusiasts were assigned to annotate the initial level of each class of BSaD. A majority voting technique [20] is used to assign the initial label of a class. An NLP expert corrected the initial labelling if any improper annotation is observed. The expert has discarded 413 texts as they had neutral and mix sentiment. To reduce bias throughout the annotation, the expert settled the labels through conversations and deliberations with the annotators [29]. Cohen's kappa [7] scores are used to estimate the inter-annotator agreement. To ensure the quality of annotation and measure the goodness of the data samples, kappa statistic is utilized [9]. It is calculated by Eq. (1) described as

$$K = \frac{p_0 - p_e}{1 - p_e}, \tag{1}$$

where p_0, p_e denotes the degree of agreement between model predictions and actual class values as if they happened by chance. We achieved a kappa score of 76.58% which indicate substantial agreement between the annotators. The final corpus contains 8122 instances of two emotion classes (2421 for positive and 5702 for negative).
- **Dataset Statistics**: After the preprocessing and annotation process, BSaD contains 8250 text documents. BSaD consists of data from various sources. Among online sources, Facebook contributes 2796, online newspapers 2306, Youtube 610, and blogs contribute 483 text documents. A substantial amount of data was collected from offline sources (2084 text documents). Table 1 shows the summary of the dataset.

Table 1 Dataset (i.e., BSaD) summary

Dataset attributes	Values
Number of documents	8122
Number of positive documents	2421
Number of negative documents	5702
Number of words	220,988
Total unique words	35,748
Maximum sentence length (in words)	233
Minimum sentence length (in words)	3
Average sentence length (in words)	27.20
File size (in bytes)	3,654,967

Figure 1 illustrates the most frequent positive and negative words of *'BSaD'* in wordclouds.[1] More highlighted words denote most frequently occur than other words in a particular class.

[1] https://www.wordclouds.com/.

(a) (b)

Fig. 1 Word cloud representation of most frequent positive and negative words

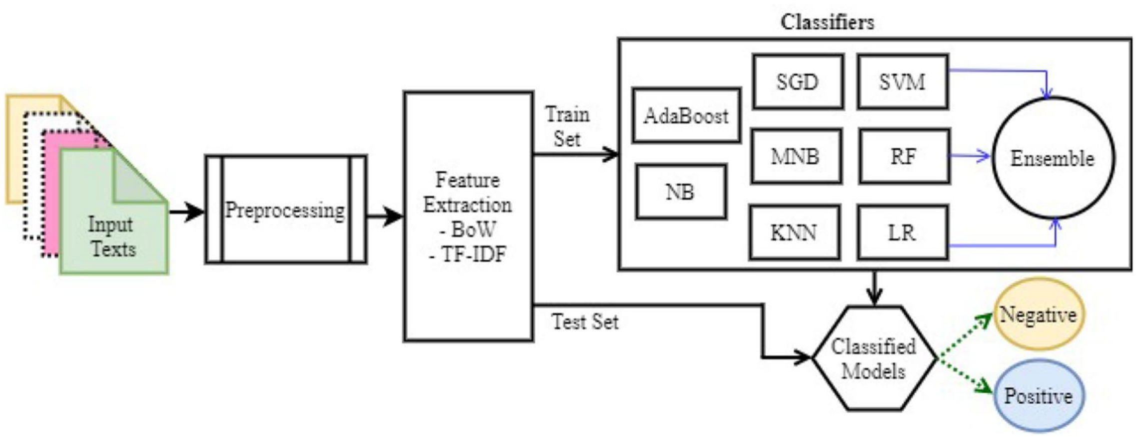

Fig. 2 Schematic diagram of the textual sentiment classification

Methodology

Figure 2 represents the schematic diagram of the proposed textual sentiment classification model. The model comprises four major modules: preprocessing, feature extraction, classifier training, and prediction. The training module utilizes eight different machine learning classifiers for system training. Among these models, SVM, RF, and LR are chosen for ensemble because of their superior performance. Finally, the trained model evaluates the test data samples to predict the category.

Preprocessing

The raw data obtained from various sources are noisy and may contain irrelevant information. Thus, a few preprocessing steps have been conducted to obtain the desired form of textual data to reduce computational complexity. Following preprocessing steps are performed by automatic filtering to develop the dataset $D[]$, where $D[]$ denotes the list of processed texts.

- Punctuation symbols, special characters, and numbers are eliminated from each text (s_i).
- Each textual content are tokenized into a set of tokens after removing the unwanted symbols or characters. A token is a string of contiguous characters grouped as a semantic unit and delimited by space, punctuation marks, and new lines.
- Words or stop words that have no contribution to decide the sentiment polarity are considered redundant and discarded. Pronoun, prefix, suffix, and conjunction are considered as stop words. Figure 3 illustrates a few resulted samples after discarding the stop words.

Fig. 3 Few examples after removing stop words

Type	Example	Raw Sentence	Processed Sentence
Pronoun	তিনি	তিনি খুব ভাল ফুটবল খেলে	খুব ভাল ফুটবল খেলে
Conjunction	এবং	পরিবেশ এবং খাওয়া দুইটাই জঘন্য	পরিবেশ খাওয়া দুইটাই জঘন্য

A processed dataset is created (i.e., BSaD) by applying the prepossessing. This step also maps textual labels into numeric labels. Numeric values 0–1 are used to represent two sentiment categories. Finally, processed texts (s_i) are stored in a dictionary indexed from $D[s_1], \dots, D[s_{8250}]$ with associated numeric labels.

Textual Feature Extraction

The machine learning models cannot interpret the textual data semantically. Thus, a mapping of words into numeric values is needed, which can be achieved by applying several feature extraction methods. This work utilizes the two most popular feature extraction methods: bag-of-word (BoW) and term frequency-inverse document frequency (TF-IDF) to extract appropriate textual features.

BoW estimates frequencies of words as features [42]. The context-relevant words which are less frequent might gain lower weights/attention than the irrelevant words with high frequency. Thus, the TF-IDF technique [37] is employed to overcome the weighting problem, which provides more weights to the contextual words. The TF (Term-frequency) and IDF (Inverse document-frequency) compute the occurrence of a word in a text and rare words in all documents, respectively. The values of TF-IDF can be estimated by Eq. (2).

$$\text{TF-IDF}(w_i, s_i) = \text{TF}(w_i, s_i) \log \frac{N}{|s\epsilon N \,:\, w\epsilon s|}. \tag{2}$$

Here, TF-IDF (w_i, s_i) indicates the value of word w_i in text document (s_i), TF (w_i, s_i) denotes the occurrence of word w_i in document (s_i), N represents the total count of text documents, and $|s\epsilon N \,:\, w\epsilon s|$ indicates the number of documents (s) contain the word (w).

In this work, the model deals with the two levels: positive and negative polarity in a sentence which is useful for determining the sentiment of the individuals. Here, we investigate the uni-gram, combination of grams: uni-gram + bi-gram and uni-gram + bi-gram + tri-gram models. After tuning the various parameters, the most suitable feature vector for BSaD is selected. Table 2 shows the shape of those feature vectors for training the machine learning models, where for uni-gram + bi-gram, max_df = 0.50 and min_df = 0.0003 and for uni-gram + bi-gram + tri-gram, max_df = 0.50 and min_df = 0.0002. The max_features is settled to 15,000.

The *n*-grams of texts are used to capture linguistic features more effectively. The *n*-gram technique considers a sequence of words to retrieve meaningful information from sentences [31]. Table 3 illustrates various *n*-gram feature representations of a sample Bengali text.

Model Training and Classification

Eight most commonly used ML classifiers (LR, RF, DT, KNN, SVM, MNB, AdaBoost, and SGD) and an ensemble of base classifiers have been prepared to investigate the performance of textual sentiment classification task. We briefly describe the basic functionality and preparation of each classifier in the following. However, readers are referred to [14, 31, 32] for in-depth studies.

- **Logistic regression (LR):** This is a linear model whose predictions are transformed by the logistic function [23]. The outcome and the cost functions of LR technique are estimated Eqs. (3) and (4), respectively

$$h_\theta(x) = \frac{1}{1 + \mathrm{e}^{-\theta^{\mathrm{T}} x}} \tag{3}$$

$$J(\Theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$
$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if: } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if: } y = 0. \end{cases} \tag{4}$$

Here, m denotes the number of training instances, $h_\theta(x^j)$ represents the hypothesis function of the jth training instance, and y^j indicates the input label of jth training instance. The 'L2' regularization technique is applied to train the logistic model and 'lbfgs' solver is used for the optimization purpose. The value of the inverse regularization strength is set to 1.

Table 2 Shape of feature vectors

N-gram feature	Feature vector
Uni-gram	6498×4233
Uni-gram + bi-gram	$6498 \times 11{,}891$
Uni-gram + bi-gram + tri-gram	$6498 \times 15{,}000$

Table 3 Representation of different n-gram features for a sample Bengali text

N-grams	“যদি লকডাউন করা হয় এই মানুষগুলো না খেয়ে মরবে ”
Uni-gram	‘যদি’, ‘ লকডাউন’, ‘করা’,‘হয়’,'এই','মানুষগুলো','না','খেয়ে','মরবে'
Bi-gram	‘যদি লকডাউন’, ‘লকডাউন করা’, ‘করা হয়’,'হয় এই','এই মানুষগুলো, 'মানুষগুলো না','না খেয়ে', 'খেয়ে মরবে'
Tri-gram	‘যদি লকডাউন করা’, ‘লকডাউন করা হয়’,'করা হয় এই','হয় এই মানুষগুলো, 'এই মানুষগুলো না', 'মানুষগুলো না খেয়ে', 'না খেয়ে মরবে'

- **Support Vector Machine (SVM):** segregates data into classes by maximizing the margin between data points and hyperplane [39]. A hyperplane can be found using Eq. (5)

$$f(y) = m^W y + c. \tag{5}$$

Here, m is a normal to the line and c is a bias. If $f(y) > 0$ then a datum belongs to one region; if $f(y) < 0$, then it belongs to another region. In our model, a ‘rbf’ kernel is used and the regularization parameter is fixed to 1. Default values of tolerance and class weight are used.

- **Random Forest (RF):** merges several decision tree to improve the predictive capability of a single DT. RF is implemented with 100 decision trees and the split ‘gini’ index is used to ensure the quality. The ‘Gini’ index is calculated by Eq. (6)

$$\text{gini} = 1 - \sum_{i=1}^{c} (y_j)^2. \tag{6}$$

Here, the total number of class and probability of jth class are denoted by c and y_j.

- **Multi-nomial Naive Bayes (MNB):** This technique used Bayes theorem to classify discrete features. According to conditional independent assumption, if the features f_1, $f_2, \dots, f_y$ are independent given a class C, the features can be calculated by bayes rule [18]. We implement MNB with a learning rate of 1 and prior probabilities are fixed based on the number of samples in a class.
- **Decision Tree (DT)**: DT comprises of external and internal nodes where classes and feature values are represented by these nodes. DT partitioned homogeneous data using entropy [Eq. (7)]

$$\text{entropy}(s) = \sum_{j=1}^{x} y_i \log_2 y_i, \tag{7}$$

where entropy(s) denotes the entropy of sample s and y_i indicates the probability of s in the training class. Measure of quality is ensured by ‘entropy’ criterion and all the features are utilized during split. Best split chosen at each node with random state 0.

- **Stochastic Gradient Descent (SGD)**: is an optimization technique that calculates gradients for a random sample rather than the whole dataset [41]. It is advantageous to train a large amount of data. Gradients are computed using Eq. (8)

$$W_{t+1} = W_t - \gamma_t \delta_w Q(z_t, w_t). \tag{8}$$

Here, W_t indicates gradient value of a randomly picked sample for iteration (t=1, 2, ..., n). γ_t denotes the learning rate and $Q(z_t, w_t)$ function minimizes the loss during training.

- **K-Nearest Neighbors (KNN)**: uses the voting technique for prediction. An instance assigned to a class having maximum votes among its k-closest neighbors [36]. These votes are determined by calculating the similarity score or distance between the sample and the intended class. For a sample (d) and class (c_i), similarity score is calculated by Eq. (9)

$$\text{score}(d, c_i) = \sum_{d_j \epsilon KNN(d)} \text{sim}(d, d_j). \tag{9}$$

Here, KNN(d) is the set of nearest neighbors of d. If d_j is a member of c_i then $(dj, ci) = 1$, otherwise 0. The d should be assigned to the class with the largest score.

- **AdaBoost**: combines several weak classifiers to create a robust classification model. It takes into account the incorrect classifications of the classifiers to assign appropriate weight to them [28]. Classifier weights are determined using Eq. (10)

$$H(x) = \left(\sum_{c=1}^{C} \alpha_c h_c(x) \right). \tag{10}$$

Here, $h_c(x)$ denotes the output of a classifier (c) for input x. The weight to the classifier is α_c which is computed by $\alpha_c = 0.5 \times \ln((1 - E)/E)$. E indicates the error rate of the classifier (c).

Ensemble Classifiers

This technique combines base classifiers to develop a specific predictive model while exploiting the individual

Table 4 Summary of classifier parameters

Classifiers	Parameters
LR	class_weight = 'balanced', max_iter = 400, random_state = 123
DT	criterion = 'entropy', random_state = 0, splitter='best'
RF	criterion = 'gini', n_estimators = 100, min_samples_split = 2
KNN	*n*_neighbors = 15, weights = 'uniform', metric = 'Minkowski', *p* = 2
SVM	probability = True, gamma = 0.0001, random_state = 0,C = 1.0, kernel = 'rbf'
MNB	additive_smoothing = 1.0, class_prior = 'None', ft_prior = 'true'
SGD	loss ="log", penalty = "l2", max_iter = 5
AdaBoost	*n*_estimators = 50, learning_rate = 1, random_state = 0

classifier's strength. Four ensemble models are developed by combining the best three base classifier models (LR, RF, and SVM). This work investigates the four ensemble techniques: (i) LR + RF, (ii) RF + SVM, (iii) LR + SVM, and (iv) LR + RF + SVM. The superior performance of the individual models might be the reason behind the success of the ensemble. Thus, we decided to use LR, RF, and SVM to design the proposed ensemble technique. Average of the probabilities of each base classifier is taken into account to decide the final label of the texts [26]. We hypothesized that the combined models of three base classifier (i.e., LR + RF + SVM) outperform all ML models and other ensemble techniques. Ensemble technique is performed by Algorithm 1.

Algorithm 1: Process of ensemble

Input: Probabilities of the classifiers
Output: Predictions of the ensemble
$E \leftarrow []$ (set of texts);
$P \leftarrow []$ (Probabilities of the classifiers);
$Pred \leftarrow []$ (list of final class);
for $s_i \epsilon S$ do
  sum = 0;
  for $j \epsilon (1, m)$ do
    $sum = P[j] + sum$;
    $j++$;
  end
  $sum = sum/m$; //normalization
  $X = argmax(sum)$;
  $Pred.append(X)$;
  i++;
end

For an emotion text (s_i) and n predefined classes $C[] = \{c_1, c_2, \dots, c_n\}$, m base classifiers provide probabilities to classify s_i. These class probabilities are summed for each instances. Ensemble method computes average of the base classifiers probabilities to classify s_i into one of the classes c_i in $C[]$. For two possible classes: positive and negative, the ensemble methods calculate two average probability values. The class with maximum probability is considered as the final label for s_i.

Experiments

Classifier models evolved in python 3.6.9 and scikit-learn 0.22.2 packages. Pandas and numpy 1.18.5 are applied to prepare the data. The 'Scikit-learn' is used to perform for ML classifiers. Parameters of the classifiers are chosen by trial-and-error procedure through empirical investigation. No prior class weight is allocated to the classes based on the number of samples. 'Gini' and 'entropy' are chosen as the criterion for RF and DT, respectively. Both classifiers employed all the features. Additive smoothing parameters fixed to 1 for MNB and 15 neighbors are used by KNN. 'Minkowski' distance metric is used where the power parameter is settled to 2. AdaBoost uses a maximum of 50

Table 5 Statistical performance measures of various ML classifiers with BoW and various TF-IDF features on BSaD

Features	Classifiers	A (%)	P (%)	R (%)	F_1 (%)
BoW	LR	77	78	77	78
	DT	72	72	72	72
	RF	78	79	78	76
	KNN	71	71	71	63
	SVM	75	74	75	73
	SGD	76	76	76	76
	MNB	79	78	79	78
	AdaBoost	76	75	76	74
TF-IDF + uni-gram	LR	79	79	79	77
	DT	72	71	72	71
	RF	79	80	79	75
	KNN	77	77	77	75
	SVM	79	79	79	79
	SGD	77	80	77	72
	MNB	75	80	75	68
	AdaBoost	76	75	76	74
TF-IDF + uni-gram + bi-gram	LR	80	81	80	77
	DT	72	71	72	72
	RF	78	80	78	75
	KNN	78	77	78	77
	SVM	80	80	80	80
	SGD	82	81	82	81
	MNB	77	81	77	72
	AdaBoost	77	76	77	75
TF-IDF + uni-gram + bi-gram + tri-gram	LR	80	82	80	78
	DT	73	73	73	73
	RF	78	80	78	75
	KNN	77	76	77	76
	SVM	81	80	81	80
	SGD	82	82	82	81
	MNB	76	81	76	71
	AdaBoost	76	75	76	74

estimators with a learning rate of 1. The SVM is implemented with the 'rbf' kernel, and the value of the regularization parameter is settled to 1. Table 4 summarised the parameters employed by the classifiers.

Results

Various statistical measures are used to evaluate the proposed textual emotion classification model, including precision (P), recall (R), accuracy (A), and F_1-score. All classifier models are implemented on the developed dataset (i.e., BSaD) for evaluation. Table 5 illustrates the performance of various ML classifiers for BoW and TF-IDF features.

Concerning BoW features, the results showed that both LR and MNB achieved the highest F_1-score of 78%. In case, TF-IDF with uni-gram features SVM gets the maximum score (79%). While for remaining two feature combination of TF-IDF, SGD outperforms other models.

Effect of Ensemble Techniques

Eight ML classifiers have been evaluated with four different types of features. After observing the performance of the classifiers, it was noticed that they obtained diverse outcomes in terms of evaluation matrices. None of the classifiers outdoes others for all feature combinations. This observation leads us to choose an ensemble approach to create a robust classifier that can outperform the baselines. This work investigates all combinations of top-performing baseline classifiers with all features combinations to ensure the diversity and accuracy of the proposed classifier. Considering the computational cost and based on classifiers F_1-score, we selected three (i.e., LR, RF, and SVM). All possible

Table 6 Evaluation results of different ensemble combinations with BoW and various TF-IDF features on BSaD

Features	Ensembles	A (%)	P (%)	R (%)	F_1 (%)
BoW	LR + RF	79	81	79	76
	LR + SVM	79	79	79	79
	RF + SVM	79	81	79	75
	LR + RF + SVM	80	80	80	80
TF-IDF +uni-gram	LR + RF	79	81	79	77
	LR + SVM	80	79	80	78
	RF + SVM	79	81	78	75
	LR + RF + SVM	81	80	81	80
TF-IDF + uni-gram + bi-gram	LR + RF	79	81	79	76
	LR + SVM	82	82	82	81
	RF + SVM	79	82	79	75
	LR + RF + SVM	80	80	80	78
TF-IDF + uni-gram + bi-gram + tri-gram	LR + RF	81	80	81	80
	LR + SVM	82	82	82	81
	RF + SVM	79	81	79	76
	LR + RF + SVM	**82**	82	82	**82**

The highest values are indicated in bold

Table 7 Evaluation results of different ensemble combinations with BoW and various TF-IDF features on DS1

Features	Ensembles	A (%)	P (%)	R (%)	F_1 (%)
BoW	LR + RF	78	80	78	77
	LR + SVM	75	76	75	75
	RF + SVM	77	80	77	77
	LR + RF + SVM	76	77	76	76
TF-IDF + uni-gram	LR + RF	78	80	78	78
	LR + SVM	75	76	75	75
	RF + SVM	77	80	77	77
	LR + RF + SVM	77	77	77	77
TF-IDF + uni-gram + bi-gram	LR + RF	80	82	80	80
	LR + SVM	79	80	79	78
	RF + SVM	79	82	79	79
	LR + RF + SVM	80	81	80	80
TF-IDF + uni-gram + bi-gram + tri-gram	LR + RF	79	82	79	79
	LR + SVM	79	80	79	79
	RF + SVM	79	82	79	79
	LR + RF + SVM	**81**	81	81	**81**

The highest values are indicated in bold

ensemble combinations have been experimented with for BoW and TF-IDF feature extraction techniques. Table 6 shows the experimental outcomes on the BSaD.

The combined features of uni-gram, bi-gram, and tri-gram (i.e., uni-gram + bi-gram + tri-gram) of TF-IDF provided the highest F_1-score (82%).

We also investigated the ensemble model's performance on two other available datasets: dataset 1 (DS1) [34] and dataset 2 (DS2) [5]. Tables 7 and 8 illustrate the outcomes of different ensemble combinations for DS1 and DS2, respectively.

Figure 4a shows the summary of the performance (regard to F_1-score) of the proposed model on three datasets concerning BoW and TF-IDF features. This analysis confirmed that the proposed ensemble technique performed better with TF-IDF than BoW features in all datasets.

Comparison with Existing Techniques

The developed ensemble model is compared with the three available methods [6, 13, 24] of sentiment classification in Bengali concerning three datasets. Figure 4b illustrates

Table 8 Evaluation results of different ensemble combinations with BoW and various TF-IDF features on DS2

Features	Ensembles	*A* (%)	*P* (%)	*R* (%)	F_1 (%)
BoW	LR + RF	72	73	72	72
	LR + SVM	69	69	69	69
	RF + SVM	71	72	71	71
	LR + RF + SVM	70	70	70	70
TF-IDF + uni-gram	LR + RF	71	73	71	71
	LR + SVM	71	71	71	71
	RF + SVM	71	72	71	70
	LR + RF + SVM	71	72	71	71
TF-IDF + uni-gram + bi-gram	LR + RF	70	71	70	69
	LR + SVM	73	73	72	72
	RF + SVM	71	73	71	71
	LR + RF + SVM	73	73	73	73
TF-IDF + uni-gram + bi-gram + tri-gram	LR + RF	73	73	73	73
	LR + SVM	73	73	73	73
	RF + SVM	73	74	72	72
	LR + RF + SVM	**74**	74	74	**74**

The highest values are indicated in bold

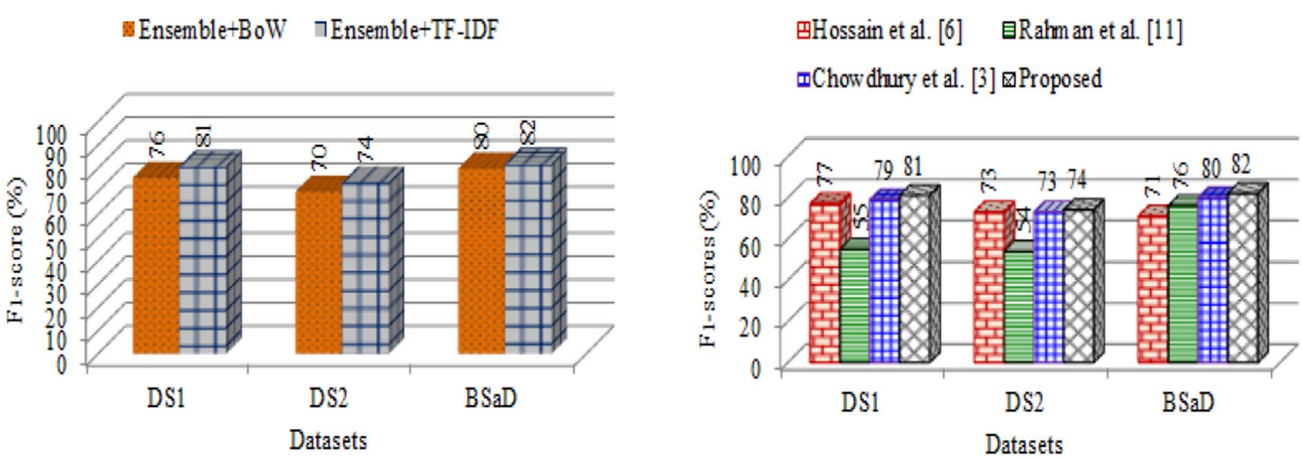

Fig. 4 Performance comparison on different datasets

the results of the comparison. The comparison results revealed that the proposed method outperformed all previous approaches with a higher F_1-score in DS1 (81%), DS2 (74%), and BSad (82%). The proposed model's ability to exploit the strength of the base classifiers helps to acquire improved performance on these datasets. Thus, the analyses confirmed that the ensemble with the TF-IDF method is better in classifying textual sentiment in Bengali.

Error Analysis

After investigating the results, it can be concluded that the proposed ensemble model is the best-performing model in classifying textual sentiment. For a better understanding of the model's performance, a detailed error analysis is performed using the confusion matrix (Fig. 5a). The confusion matrices of the existing methods [6, 13, 24] on BSaD are also presented in Fig. 5b–d.

Among the existing models, Chowdhury et al. [6] obtained the maximum accuracy. Approximately 93% of negative class instances are classified correctly for the proposed model, while only 58.85% of positive instances are accurately classified. Among 503 positive test instances, 207 instances are wrongly categorized as negative. On the other hand, only 80 examples were classified as positive among 1122 negative data. The limited number of positive training data might be the reason behind this biased performance. Moreover, the proposed model cannot capture the semantic information of the texts, which is crucial to identify the sentiment. Therefore, more sophisticated feature extraction

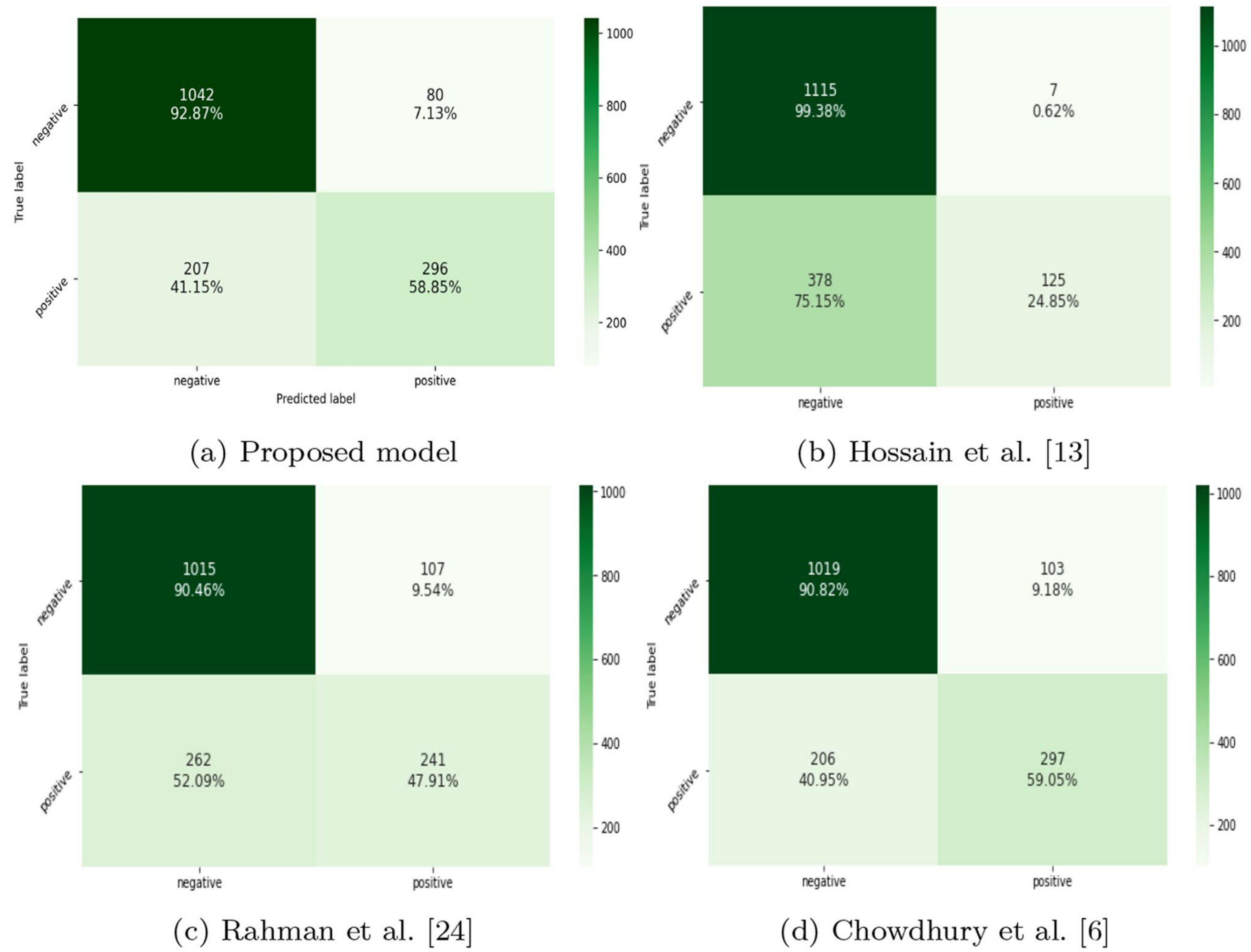

Fig. 5 Confusion matrix of the proposed and existing techniques on BSaD

techniques can be investigated with more diverse data to improve the system's predictive accuracy.

Conclusion

This paper investigated the various ML techniques for classifying the textual sentiment in Bengali into two classes: positive and negative. Eight widely used ML techniques and one ensemble technique (using LF, RF, and SVM) have been implemented with tuned parameters. The performance of the models has been investigated on the developed dataset (BSaD) and two benchmark datasets. The experimental outcomes revealed that the ensemble method with TF-IDF (unigram + bi-gram + tri-gram) features outperformed the other classifier models with the highest accuracy (82%) on the developed dataset in classifying textual sentiment in Bengali. The performance of the current implementation can be enhanced by consolidating more numerous textual data in BSaD with multiple domains. Furthermore, text with mixed or neutral sentiment and emojis can be analyzed to improve the model's generalization capacity.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Akhtar MS, Ekbal A, Cambria E. How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble [application notes]. Comput Intell Mag. 2020;15(1):64–75. https://doi.org/10.1109/MCI.2019.2954667.
2. Akhtar MS, Gupta D, Ekbal A, Bhattacharyya P. Feature selection and ensemble construction. Knowl Based Syst. 2017;125(C):116–35. https://doi.org/10.1016/j.knosys.2017.03.020.

3. Amrani YA, Lazaar M, Kadiria KEE. Random forest and support vector machine based hybrid approach to sentiment analysis. Procedia Comput Sci. 2018;127:511–20.
4. Bakar A, Razi MF, Norisma I, Liyana S, Norazlina K. Sentiment analysis of noisy Malay text: state of art, challenges and future work. IEEE Access. 2020;8:24687–96.
5. Banglapedia: Bangla language. 2019. https://www.kaggle.com/tazimhoque/bengali-sentiment-text. Accessed 23 Mar 2020.
6. Chowdhury RR, Hossain MS, Hossain S, Andersson K. Analyzing sentiment of movie reviews in Bangla by applying machine learning techniques. In: International conference on Bangla speech and language processing (ICBSLP). IEEE; 2019. p. 1–6.
7. Cohen J. A coefficient of agreement for nominal scales. Educ Psychol Meas. 1960;20(1):37–46.
8. Das A, Iqbal MA, Sharif O, Hoque MM. BEmoD: development of Bengali emotion dataset for classifying expressions of emotion in texts. In: Intelligent computing and optimization. ICO 2020. Advances in intelligent systems and computing, vol. 1324. Berlin: Springer; 2021. p. 1124–36.
9. Das A, Sharif O, Hoque MM, Sarker IH. Emotion classification in a resource constrained language using transformer-based approach. In: Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: student research workshop. Association for Computational Linguistics; 2021. p. 150–8 (Online). https://doi.org/10.18653/v1/2021.naacl-srw.19. https://aclanthology.org/2021.naacl-srw.19
10. Dashtipour K, Ieracitano C, Morabito FC, Raza A, Hussain A. An ensemble based classification approach for persian sentiment analysis. In: Progresses in artificial intelligence and neural systems. Singapore: Springer; 2021. p. 207–15.
11. Gamal D, Alfonse M, El-Horbaty ESM, Salem ABM. Analysis of machine learning algorithms for opinion mining in different domains. Mach Learn Knowl Extr. 2019;1(1):224–34.
12. Garg K, Lobiyal DK. Hindi EmotionNet: a scalable emotion lexicon for sentiment classification of Hindi text. ACM Trans Asian Low Resour Lang Inf Process. 2020;19(4):1–35.
13. Hossain E, Sharif O, Hoque MM. Sentiment polarity detection on Bengali book reviews using multinomial naive Bayes. 2020. arXiv preprint arXiv:2007.02758.
14. Hossain E, Sharif O, Hoque MM. NLP-CUET@LT-EDI-EACL2021: multilingual code-mixed hope speech detection using cross-lingual representation learner. In: Proceedings of the first workshop on language technology for equality, diversity and inclusion. Kyiv: Association for Computational Linguistics; 2021. p. 168–74. https://aclanthology.org/2021.ltedi-1.25.
15. Hossain E, Sharif O, Hoque MM, Sarker IH. SentiLSTM: a deep learning approach for sentiment analysis of restaurant reviews. 2020. arXiv preprint arXiv:2011.09684.
16. Islam MS, Islam MA, Hossain MA, Dey JJ. Supervised approach of sentimentality extraction from Bengali facebook status. In: 2016 19th International conference on computer and information technology (ICCIT). IEEE; 2016. p. 383–7.
17. Lai Y, Zhang L, Han D, Zhou R, Wang G. Fine-grained emotion classification of Chinese microblogs based on graph convolution networks. World Wide Web. 2020;23(5):2771–87.
18. Le CC, Prasad P, Alsadoon A, Pham L, Elchouemi A. Text classification: Naïve Bayes classifier with sentiment lexicon. IAENG Int J Comput Sci. 2019;46(2):141–8.
19. Luo L. Network text sentiment analysis method combining LDA text representation and GRU-CNN. Pers Ubiquitous Comput. 2019;23:405–12.
20. Magatti D, Calegari S, Ciucci D, Stella F. Automatic labeling of topics. In: 2009 Ninth international conference on intelligent systems design and applications. IEEE; 2009. p. 1227–32.
21. Mamta AE, Bhattacharyya P, Srivastava S, Kumar A, Saha T. Multi-domain tweet corpora for sentiment analysis: resource creation and evaluation. In: Proceedings of the 12th LREC. Marseille: European Language Resources Association; 2020. p. 5046–54.
22. Prabowo R, Thelwall M. Sentiment analysis: a combined approach. J Informetr. 2009;3(2):143–57.
23. Pranckevičius T, Marcinkevičius V. Application of logistic regression with part-of-the-speech tagging for multi-class text classification. In: 2016 IEEE 4th workshop on advances in information, electronic and electrical engineering (AIEEE). IEEE; 2016. p. 1–5.
24. Rahman M, Kumar Dey E, et al. Datasets for aspect-based sentiment analysis in Bangla and its baseline evaluation. Data. 2018;3(2):15.
25. Sarkar K. Sentiment polarity detection in Bengali tweets using LSTM recurrent neural networks. In: 2019 Second international conference on advanced computational and communication paradigms (ICACCP). IEEE; 2019. p. 1–6.
26. Sarkar K. Heterogeneous classifier ensemble for sentiment analysis of Bengali and Hindi tweets. Sādhanā. 2020;45(1):1–17.
27. Sarkar K, Bhowmick M. Sentiment polarity detection in Bengali tweets using multinomial naïve Bayes and support vector machines. In: 2017 IEEE Calcutta conference (CALCON). IEEE; 2017. p. 31–6.
28. Schapire RE. Explaining adaboost. 2013. https://doi.org/10.1007/978-3-642-41136-6_5.
29. Sharif O, Hoque MM. Identification and classification of textual aggression in social media: resource creation and evaluation. In: Chakraborty T, Shu K, Bernard HR, Liu H, Akhtar MS, editors. Combating online hostile posts in regional languages during emergency situation. Cham: Springer; 2021. p. 9–20.
30. Sharif O, Hoque MM, Hossain E. Sentiment analysis of Bengali texts on online restaurant reviews using multinomial naïve Bayes. In: International conference on advances in science, engineering and robotics technology (ICASERT). IEEE; 2019. p. 1–6.
31. Sharif O, Hoque MM, Kayes ASM, Nowrozy R, Sarker IH. Detecting suspicious texts using machine learning techniques. Appl Sci. 2020;10(18). https://doi.org/10.3390/app10186527.
32. Sharif O, Hossain E, Hoque MM. Combating hostility: Covid-19 fake news and hostile post detection in social media. 2021. arXiv preprint arXiv:2101.03291.
33. Tabassum N, Khan MI. Design an empirical framework for sentiment analysis from Bangla text using machine learning. In: Proceedings of ECCE. IEEE; 2019. p. 1–5.
34. Taher S, Akhter K, Hasan KM. Bangla dataset for opinion mining. 2018. https://doi.org/10.13140/RG.2.2.20214.96327.
35. Taher SA, Akhter KA, Hasan KA. N-gram based sentiment mining for Bangla text using support vector machine. In: 2018 International conference on Bangla speech and language processing (ICBSLP). IEEE; 2018. p. 1–5.
36. Tan S. An effective refinement strategy for KNN text classifier. Expert Syst Appl. 2006;30(2):290–8. https://doi.org/10.1016/j.eswa.2005.07.019.
37. Tokunaga T, Makoto I. Text categorization based on weighted inverse document frequency. In: Special interest groups and information process Society of Japan (SIG-IPSJ). Citeseer; 1994.
38. Wahid MF, Hasan MJ, Alom MS. Cricket sentiment analysis from Bangla text using recurrent neural network with long short term memory model. In: International conference on Bangla speech and language processing (ICBSLP). IEEE; 2019. p. 1–4.
39. Xia H, Yang Y, Pan X, Zhang Z, An W. Sentiment analysis for online reviews using conditional random fields and support vector machines. Electron Commer Res. 2020;20(2):343–60.

40. Xu G, Yu Z, Yao H, Li F, Meng Y, Wu X. Chinese text sentiment analysis based on extended sentiment dictionary. IEEE Access. 2019;7:43749–62.
41. Zhang T. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In: Proceedings of the twenty-first international conference on machine learning, ICML '04. New York: Association for Computing Machinery; 2004. p. 116. https://doi.org/10.1145/1015330.1015332.
42. Zhang Y, Jin R, Zhou ZH. Understanding bag-of-words model: a statistical framework. Int J Mach Learn Cybern. 2010;1(1–4):43–52.