*Sādhanā* (2020)45:196
https://doi.org/10.1007/s12046-020-01424-z

© Indian Academy of Sciences

# Heterogeneous classifier ensemble for sentiment analysis of Bengali and Hindi tweets

KAMAL SARKAR

Department of Computer Science and Engineering, Jadavpur University, Kolkata, India
e-mail: jukamal2001@yahoo.com

MS received 24 July 2019; revised 12 January 2020; accepted 25 April 2020

**Abstract.** Sentiment analysis is an essential step for analysing social media texts such as tweets and other posts on the various micro-blogging sites. The basic step of sentiment analysis is sentiment polarity detection, which identifies whether an input piece of social media text is positive, negative or neutral. In this paper, we present an approach that combines heterogeneous classifiers in an ensemble for sentiment polarity detection in Bengali and Hindi tweets. Our proposed method constructs an ensemble of three different base classifiers where the feature set for each base classifier is different from each other. We have also incorporated an external knowledge base called sentiment lexicon to augment tweet words with sentiment polarity information retrieved from the sentiment lexicon. Experimental results show the effectiveness of our proposed heterogeneous ensemble model for sentiment polarity detection for both Bengali and Hindi languages. It has been shown that our system outperforms other existing Bengali and Hindi sentiment classification systems to which it is compared.

**Keywords.** Bengali tweets; hindi tweets; sentiment polarity detection; machine learning; ensemble; classifier combination; deep learning.

## 1. Introduction

At the present time, a massive amount of social media and user-generated content is becoming available on the internet. The social media texts can be of various types—blog posts, tweets, opinions and comments, which can pertain to the different domains such as sports, crickets, politics, music and movies. This vast amount of online social media data can be used for deriving market intelligence through evaluation of public opinions and views. Knowledge extracted from the social media data can also be useful in social and political policy making as well as enriching diverse academic fields such as political science, psychology and sociology.

Since manual processing of the vast amount of social media texts is a difficult task, we need an efficient and accurate system that can analyse and summarize the opinions coming from various online sources. One of the important steps for analysing social media texts is to automatically classify polarity of sentiment expressed in a comment or tweet. This area of research is known as sentiment analysis though two research areas—sentiment analysis and opinion mining—which are closely related to each other. Sentiment is expressed in the form of an opinion, a subjective impression, a thought or judgment prompted by feelings [1]. It also includes emotions.

Our main focus of our work is to find solution to the problem of sentiment analysis that deals with classifying sentiment polarity of a tweet into positive, negative or neutral. We have carried out our study on sentiment analysis for Bengali and Hindi tweets.

Though sentiment analysis is one of the hot research topics and many researchers have already published their research works in this area, the most existing works on sentiment analysis deal with sentiment classification in English language domain. There are a limited number of published research works on sentiment analysis for Indian languages (SAIL) and most existing works on Indian language sentiment analysis either use lexicon-based approaches or use a single classifier with a richer feature set. Since our study includes sentiment analysis of tweets for two Indian languages—Bengali and Hindi, which are both morphologically rich languages, it is difficult to find a better solution to our problem using the single classifier, which may suffer from data sparseness problem caused by morphological variants of words and over-fitting problem. We think that character *n*-gram features can be useful in dealing with this kind of data sparseness problem and the classifier ensemble model can deal with over-fitting problem to some extent. Motivated by this fact, we have developed a system using heterogeneous classifier ensemble method for sentiment polarity detection in Bengali and Hindi tweets. This approach first constructs three different

Published online: 06 August 2020

base models. The first base model uses Multinomial Naïve Bayes classifier with traditional word *n*-gram features and sentiment lexicon that consists of a collection of positive, negative and neutral polarity words, the second base model also uses Multinomial Naïve Bayes classifier, but the feature set is different. The character *n*-gram features along with sentiment lexicon are used for developing the second base model and the third base model uses support vector machine (SVM) with linear kernel, unigram and sentiment lexicon features.

In the heterogeneous classifier ensemble method, the predictions of the different base models are combined into a single model for improving accuracy. The basic difference between homogeneous and heterogeneous ensemble methods is as follows. In the homogeneous ensemble method, the base models are generated using the same base classifier, the same feature set and creating the training set for each individual base model by random sampling the original training dataset. On the other hand, in the heterogeneous ensemble method, the base models can be generated in either of the following two ways: (1) using different classifiers, the same feature set and creating the training set for each individual base model by random sampling the original training dataset or (2) using different classifiers, different feature sets and creating the training set for each individual base model by representing the original training dataset with the corresponding feature set. The heterogeneous classifier ensemble method, which we have used for developing our proposed system, combines three base models where each base model uses a feature set different from others. Since the feature set is different, the representation of training data becomes different for each base model though the same training corpus is used for developing each base model. The contributions of the work are summarized as follows:

1. We discuss the merits of using heterogeneous classifier ensemble for performing sentiment analysis of tweets written in two different Indian languages—Bengali and Hindi.
2. We prove the effectiveness of heterogeneous classifier ensemble for Indian language sentiment analysis by comparing the performance of the proposed approach to those of deep learning models—Long Short-Term memory (LSTM), Bidirectional (BILSTM) and Convolutional Neural Network (CNN).
3. We show the usefulness of our proposed approach for sentiment analysis for two different Indian languages—Bengali and Hindi.

The organization of the rest of the paper is as follows. Section 2 presents previous works related to our work. In section 3, we have described the proposed methodology including the details of base classifiers, feature extraction and the ensemble technique. Section 4 presents description of datasets, experimental results and comparisons of the proposed method to some existing methods. Section 5 concludes the paper.

## 2. Related work

The earlier works on sentiment analysis use computational linguistics, natural language processing (NLP) and text mining techniques [2], which require analysis of deeper linguistic knowledge [10–13]. Such techniques also use sentiment lexicon. The approach [18] that uses sentiment lexicon for sentiment polarity classification depends solely on the knowledge base called sentiment lexicon, which is usually constructed by manually collecting the polarity (positive, negative and neutral) terms [19]. A sentiment lexicon can also be constructed using some automatic processes [10, 20–24]. A kind of sentiment lexicon called SentiWordNet [25] is constructed through a manual process by analysing the glosses retrieved from WordNet [26]. Though the sentiment lexicons have been used in many sentiment analysis approaches the main problem with sentiment-lexicon-based approach is to develop and maintain the sentiment lexicons, which vary from one domain to another domain and one language to another language.

To overcome this situation, various machine-learning-based techniques are used for sentiment polarity detection [2–9]. Most researchers prefer to use supervised machine learning algorithms for sentiment analysis because the machine learning algorithms can be easily and quickly trained with the new training data to port them to a new domain. When supervised machine learning algorithms are used for sentiment polarity detection they have to be trained with the data prepared in a certain form called feature vectors, which are created by converting each text item from a sentiment-polarity-labelled corpus into a feature vector. The obtained feature vectors need to be labelled by the labels of the corresponding text items before they are submitted to the machine learning algorithm. The features that are proven to be effective for sentiment polarity classification are word *n*-grams, words occurring in the context, punctuation, etc. The supervised machine learning algorithms that are usually preferred by the researchers are SVM, Naïve Bayes, Decision Tree, *K*-nearest Neighbour (KNN) and Artificial Neural Networks (ANN) [16, 18, 27–29].

Another machine learning technique called ensemble learning has been widely used in many areas for solving classification problems. Some previous research works on sentiment analysis of English tweets have reported the efficacy of ensemble classifier models in sentiment classification tasks. Onan *et al* [30] proposed an ensemble classifier model that used a multi-objective differential evolution algorithm for selecting optimal number of base classifiers. They compared weighted and un-weighted voting schemes for constructing an ensemble of classifiers. da

Silva *et al* [31] proposed an ensemble classifier based on majority vote for Twitter sentiment analysis. Rodríguez-Penagos *et al* [32] used SVMs and Conditional Random Fields as base learners to form an ensemble of classifiers. Hassan *et al* [33] developed an ensemble technique using bootstrap aggregation techniques. They also proposed an algorithm that would select the most appropriate classifier among all the base classifiers. Ankit and Saleena [34] presented an ensemble method for Twitter sentiment analysis that used Naïve Bayes classifier, Random Forest classifier, SVMs and Logistic Regression as the base classifiers.

Though research works on sentiment polarity detection have been carried out in the different genres such as blogs [14], discussion boards or forums [15], user reviews [16] and expert reviews [17], many previous research works have concentrated on sentiment polarity detection in texts written in English language. However, since Indian social media texts are multilingual in nature, we also need a system that can detect the sentiments of social media texts written in Indian languages. To fill up the gap, a contest on SAIL Tweets was conducted in 2015. It was co-located with MIKE 2015 conference held at IIIT, Hyderabad, India [35]. The two most spoken Indian languages—Bengali and Hindi—were also included in this shared task. In addition to this shared task, some researchers have recently published several research papers on sentiment analysis for Bengali (Bangla) [29, 36–39].

The Bengali sentiment polarity detection models presented in [36] and [38] both use Multinomial Naïve Bayes classifier, but the feature set is not the same. The model presented in [36] uses sentiment lexicon and word *n*-gram features whereas the model presented in [38] uses sentiment lexicon and character *n*-gram features. The authors of [36] have also compared in this paper the performance of Multinomial Naïve Bayes classifier to that of an SVM-based system, which used sentiment lexicon and word unigram features. Though these three models are similar in some way to the first, the second and the third base models that our proposed ensemble model comprises, we have used more optimized feature sets while constructing our base models whereas the existing systems presented in [36, 38] do not use the optimized feature sets.

Like Bengali language, we also find limited research works on sentiment analysis for Hindi language. The work presented in [40] describes sentiment analysis for Hindi language using unsupervised lexicon method, whereas the work presented in [41] uses a fall-back strategy and SVM classifier for sentiment analysis of Hindi reviews. Another approach to sentiment analysis of Hindi reviews presented in [42] uses Hindi sentiment lexicon and Negation and Discourse rules for sentiment classification.

In recent years, deep learning techniques such as LSTM, BILSTM and CNN are being applied to sentiment analysis tasks [37, 43, 44]. A deep learning model for Bengali tweet sentiment analysis has been presented in [39] and [37]. The approach presented in [39] used recurrent neural networks called LSTM for model development and the approach used in [37] used CNN for sentiment analysis of Bengali tweets.

The work presented in [45] used CNN for sentiment analysis of Hindi reviews. An approach based on deep recurrent neural networks to sentiment analysis of Punjabi texts describing suicidal cases has been presented in [46]. The main purpose of the work in [46] was to predict intensity of negative sentiment score along with the class of the suicide case. We found limited research works on sentiment analysis of Hindi tweets that use deep learning models, though one interesting work that combines CNN and SVM for sentiment analysis of Hindi tweets was found in [47].

## 3. Proposed methodology

The proposed system uses heterogeneous ensemble model for sentiment polarity detection in Bengali and Hindi tweets. In this section, we discuss the major steps for our proposed approach: (1) data cleaning and preprocessing, (2) base classifiers and features and (3) ensemble model development and sentiment polarity classification.

### 3.1 *Data cleaning and preprocessing*

In this step, the entire data collection is processed to remove irrelevant characters: ”/”, ”“ „ ”“-“”, ”“-, ”“/, ”“(“”, ”“), ”“@, ”“#,”””, ””’, “”.”nd “”*””. This is important for tweet data because tweet data is noisy. Words that occur only once in the corpus are also removed from the tweets as irrelevant features.

### 3.2 *Base classifiers and features*

In our proposed heterogeneous ensemble method, we combine outputs of several base classifiers that are heterogeneous in nature. Here each base classifier uses a different set of features.

The base classifiers are combined using the non-trainable classifier combination strategy. The most common such classifier combination strategies that we have tried for our implementation are majority voting, average of probability and max rule. Out of these three classifier combination strategies, the best strategy has been chosen through experimentation. Details of classifier combination strategies will be discussed later in this section.

Architecture of our proposed ensemble model is shown in Figure 1. Our proposed model is basically a hybrid classification model, which is implemented by developing the base classification models first and then combining the predictions of the developed base classifiers. Here the base classifiers are heterogeneous in the sense that each base

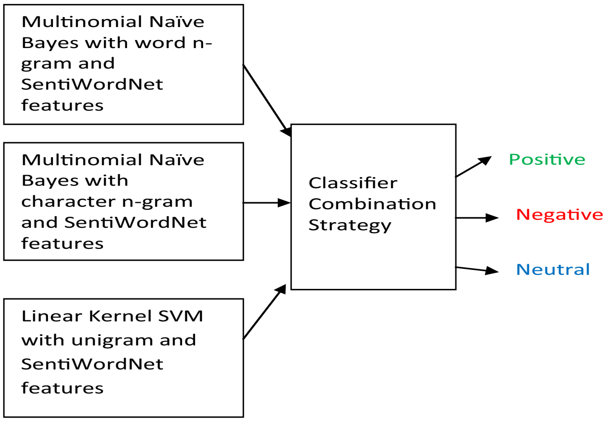

Multinomial Naïve Bayes with word n-gram and SentiWordNet features
Multinomial Naïve Bayes with character n-gram and SentiWordNet features
Linear Kernel SVM with unigram and SentiWordNet features
Classifier Combination Strategy
Positive
Negative
Neutral

**Figure 1.** System architecture of the heterogeneous ensemble method applied to Bengali and Hindi tweet sentiment analysis

classifier uses a different set of features. The base classifiers that we have incorporated in our proposed ensemble model are (1) Multinomial Naïve Bayes with sentiment lexicon and word *n*-gram features, (2) Multinomial Naïve Bayes with sentiment lexicon and character *n*-gram features and (3) SVM with sentiment lexicon and unigram features. Here, sentiment lexicon is an external knowledge base consisting of collection of manually classified polarity words [49].

In the subsequent sub-sections, we will discuss the details of the various base classifiers along with their feature sets. We will also discuss how the ensemble of classifiers is developed by combining the base classifiers using various classifier combination strategies such as majority voting, average of probability and max rule.

### 3.2.1 *Base classifier 1: Multinomial Naïve Bayes with sentiment lexicon and word n-gram features*

The first base classifier used in our proposed ensemble method uses Multinomial Naïve Bayes [36, 38] classifier with word *n*-gram features, which works as follows.

For feature extraction, the word *n*-gram tokenizer is used to tokenize the input tweet into word *n*-grams such as unigram, bigram, trigram, etc. Each word *n*-gram is considered as a feature. Frequency of a word *n*-gram is considered as the feature value. The number of features is selected as per class basis. Since the most frequent *k* features are selected per class, the feature set for the classifier is formed by taking union of features selected for the different classes. In case of a tie, all features with the same frequency are included in the feature set. For implementation of this base classifier, the value of *k* is determined through experimentation. Thus a vocabulary *V* is created with the set of selected word *n*-grams. Then a tweet $T_i$ is represented using a bag-of-word *n*-grams model that represents the tweet as a feature vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{it})$, where $x_{it}$ is the number of times the vocabulary term $w_t$ occurs in the tweet $T_i$ and $t = 1, \ldots, |V|$.

The Multinomial Naïve Bayes classifier classifies a tweet $T_i$ based on the posterior probability for a sentiment class given the tweet. The posterior probability for a sentiment class given the tweet is calculated as follows:

$$P(C|T_i) \propto P(C) \prod_{t=1}^{|V|} P(w_t|C)^{x_{it}} \tag{1}$$

where |V| is the size of vocabulary, $W_t$ is the *t*-th term (word *n*-gram) in the vocabulary, *C* is a sentiment label or class, $P(C)$ is the prior probability of the class *C* and $X_{it}$ is the count of occurrences of the vocabulary term $w_t$ in the tweet $T_i$.

Since many vocabulary words do not occur in the tweet, for most cases, we will see that $x_{it} = 0$. Hence, for the vocabulary terms for which $x_{it} = 0$, the value of the term $P(w_t|C)^{x_{it}}$ becomes 1. Hence equation (1) can be modified as follows:

$$P(C|T_i) \propto P(C) \prod_{j=1}^{m} P(t_{ij}|C) \tag{2}$$

where the tweet $T_i$ is a sequence of terms, $T_i = (t_{i1}, t_{i2}, \ldots, t_{im})$, $P(C|T_i)$ is the probability that the tweet $T_i$ is in the sentiment class *C*, $P(t_{ij}|C)$ is the probability that the *j*-th term (word *n*-gram) of the tweet $T_i$ belongs to the class *C*, $P(C)$ is the prior probability of the class *C* estimated from the training data and *m* is the total count of terms contained in the tweet $T_i$.

According to equation (2), the probability that a tweet belongs to a sentiment class *C* is calculated based on the product of term likelihoods and the prior probability of the class *C*. Multinomial Naïve Bayes that we have applied to our sentiment analysis task has been discussed in [36, 38] as a part of our previous work. For implementation of this base model, we have considered word *n*-grams as the primary features and we have taken word *n*-grams up to trigrams; that is, from each tweet, the terms such as unigrams ($n = 1$), bigrams ($n = 2$) and trigrams ($n = 3$) are generated to consider them as the features. Along with the word *n*-gram features, we have also utilized sentiment polarity information extracted from the sentiment lexicon. Given a tweet word, the corresponding sentiment polarity information is retrieved and incorporated in our Bayesian model. The sentiment lexicon that we have used for our task has been created by collecting sentiment polarity words from SentiWordNet for Indian Languages[1] [49]. Thus our used sentiment lexicon is a collection of positive, negative and neutral words for a particular language (Bengali or Hindi). In order to incorporate sentiment lexicon in our base models, we have used a tweet augmentation method that augments each tweet word with a special pseudo-word

[1] http://amitavadas.com/sentiwordnet.php

"#Pos" if searching with the tweet word in the sentiment lexicon returns "positive", "#Neg" if searching with the tweet word in the sentiment lexicon returns "negative" and "#Neu" if searching with the tweet word in the sentiment lexicon returns "neutral". For example, the tweet "(a very beautiful movie, a great movie)" is augmented as follows: "(a very #Pos beautiful #Pos movie, a great #Pos movie)". With this new augmentation, the formula for computing posterior probability is modified as follows:

$$P(C|T_i) \propto$$
$$P(C)\left[\prod_{j=1}^{m} P(t_{ij}|C)\right] P(\#Pos|C)^{m_1} P(\#Neg|C)^{m_2} P(\#Neu|C)^{m_3} \tag{3}$$

where $m$ is the total count of tokens (word $n$-grams) in the tweet, $m_1$ is the number of positive words contained in the tweet (a tweet word is considered as positive when sentiment lexicon says that it is positive) , $m_2$ is the number of negative words contained in the tweet, $m_3$ is the number of neutral words contained in the tweet, $P(\#Pos|C)$ is the probability that a tweet word with positive polarity belongs to the sentiment class $C$, $P(\#Neg|C)$ is probability that the tweet word with negative polarity belongs to the class $C$ and $P(\#Neu|C)$ is the probability that the word with neutral polarity belongs to the class $C$.

Equation (3) gives us a new method for computing the posterior probability for a tweet. In this method, the posterior probability is affected by the number of each type of polarity words the tweet contains. For example, if a tweet contains relatively more number of negative words compared with the other two polarity type words, the negative polarity of the tweet is boosted and its probability of being in negative class is relatively increased compared with other two sentiment classes.

#### 3.2.2 *Base classifier 2: Multinomial Naïve Bayes with sentiment lexicon and character n-gram features*

This base classification model also works based on the same principle as the first base model described in the previous sub-section. The only difference of this classifier from the earlier one is the type of features used. Here we have used character $n$-gram features whereas, in the previous case, we have used word $n$-gram features. The difference between character $n$-grams and word $n$-grams is illustrated with an example given here.

For the input text: Ki Darun food!", the word $n$-grams (for $n = 1 - 2$) are: "Ki", "Darun", "food", "Ki Darun", "Darun food" whereas the character $n$-grams for $n = 4$ are "Ki D", "i Da", " Dar", "Daru", "arun", "run " , "un f", "n fo", " foo", "food", "ood!".

One of the advantages of using character $n$-gram features is that it helps alleviate out-of-vocabulary problem, which occurs when the word occurring in the test tweet is absent in the training data. This problem is also known as data sparseness problem. For this base classifier, a character $n$-gram is considered as a feature and its frequency in the tweet is considered as the feature value. For best performance of the classifier, we have considered character $n$-grams of length varying from 2 to 5 (i.e., $n = 2 - 5$). The posterior probability for a tweet represented as a set of character $n$-grams can be calculated by adapting equation (3) to character $n$-gram features. Vocabulary is created using a method similar to that described in the earlier subsections (i.e., vocabulary is created by taking union of the features selected for the different classes).

#### 3.2.3 *Base classifier 3: SVMs with sentiment lexicon and word unigram features*

Since SVM [50] has the inherent capability to deal with high-dimensional data, we have used SVM as the third base classifier for our proposed ensemble model. Since prior research shows that the linear kernel is more useful in text classification task [51, 52], we have used SVM with linear kernel for implementing this base classifier. Before feature extraction, the sentiment-lexicon-based tweet augmentation strategy as mentioned in the earlier section has also been used. For this base classifier, unigrams are considered as the features, the frequency of a unigram in the tweet is considered as its feature value and the bag-of-words model is also used to represent each tweet as a feature vector. For achieving better performance, all unigrams are not taken as features. Based on word frequency statistics, the $k$ most frequent unigrams per class are also selected as the features for developing this base classifier. Finally a tweet is represented as a feature vector using the method as described earlier in this paper. Feature vector representing each training tweet is labelled with the label of the corresponding training tweet.

### 3.3 *Heterogeneous classifier ensemble for sentiment polarity classification*

For overall model development, three different base classifiers discussed in the earlier sub-sections are combined using a model combination strategy. Our developed model learns from the training data how to classify a tweet into one of three sentiment polarity classes—positive, negative and neutral. During testing phase, the unlabelled tweet is presented for classification to the trained model. The label of the test tweet, assigned by the model, is considered as the sentiment label of the corresponding tweet. While developing the base models, we have chosen the best possible configuration of the model parameters through 10-fold cross-validation method.

Though there are a number of non-trainable model combination strategies [48], we observe that the three model combination strategies discussed here are more useful for solving our sentiment classification problem.

3.3.1 *Majority voting* According to this classifier combination strategy, an input tweet is assigned the class label that receives the largest number of votes cast by the base classifiers. The class that receives the largest number of votes is then selected as the consensus (majority) decision. Here the class label $C$ is said to receive a vote from the classifier $X$ when the classifier $X$ predicts the class label $C$ for an input tweet $T$. For an example, given a test tweet when two base classifiers predict the sentiment class for the test tweet as "positive" and one base classifier predicts the sentiment class of the tweet as "negative" , we say that positive class has received two votes and the negative class has received one vote. Since the positive class receives the largest number of votes, the test tweet is assigned the label "positive".

3.3.2 *Average of probabilities* Unlike majority voting rule, which uses hard class labels predicted by the base classifiers, this classifier combination rule uses the soft class label information. Given the test input tweet $T$, for any class $C$, $m$ base classifiers give $m$ probability values (confidence values) as the output in which each probability value indicates confidence score of a base classifier in classifying the input tweet as the class $C$. The average of $m$ probability values given by $m$ base classifiers while predicting the class $C$ is taken as the confidence value of the ensemble model in classifying the input tweet as the class $C$. Thus, for three possible classes—positive, negative and neutral, the ensemble model computes three average probability values. Out of these three classes, the class for which the average probability value is the maximum is considered as the class of the test tweet.

3.3.3 *Max rule* This classifier combination rule also combines the soft predictions of the $m$ base classifiers. Instead of averaging $m$ probability values given by $m$ base classifiers for any class $C$, the max rule computes the maximum of the $m$ probability values and considers it as the confidence value of the ensemble model in classifying the input tweet as the class $C$. Thus, for each of the possible classes—positive, negative and neutral, the ensemble model outputs one confidence score. Out of these three classes, the class for which the confidence score is the maximum is considered as the class of the input tweet.

## 4. Evaluation and experimental results

### 4.1 *Datasets*

We have conducted several experiments using Bengali and Hindi training datasets released for a shared task on SAIL Tweets, held in 2015. This shared task was co-located with conference MIKE 2015 held at IIIT Hyderabad, India [35]. Table 1 shows a summary of the datasets that we have used for our experimentation.

**Table 1.** SAIL 2015 datasets for Bengali and Hindi sentiment analysis

| Languages | Data | Number of tweets |
|---|---|---|
| Bengali | Training | 1,000 |
| | Test | 500 |
| Hindi | Training | 1,293 |
| | Test | 467 |

### 4.2 *Experiments and results*

We have developed two different language-specific ensemble models—one for Bengali language and another for Hindi language. We have used the Weka machine learning workbench[2] for implementing our models. To prove effectiveness of our proposed ensemble model for both the languages, the type of features and the base classifiers used for Bengali sentiment analysis are also used for Hindi sentiment analysis. The number of base classifiers is also the same for the two languages. The input data and the parameter settings are the only difference for these two different ensemble models—one for Bengali language and another for Hindi language. The performance of each developed model is measured in terms of accuracy. For model evaluation and comparison, 10-fold cross-validation performance has been considered. For Bengali language, we have combined the SAIL 2015 Bengali training and test data to form a dataset consisting of 1500 tweets and the average accuracy over 10 folds is computed for each model. For Hindi language, SAIL 2015 Hindi training and test data are also combined to form a dataset consisting of 1760 tweets. The model performance on Hindi dataset has also been obtained by averaging accuracy of the model over 10 folds. The obtained results achieved by the two language-specific models have been reported in this paper. We conducted several experiments to find the best configuration of each base classifier and to decide which combination rule is the best to form the ensemble of heterogeneous base classifiers. In the subsequent sub-sections, for each language domain, we will separately discuss the performance of each individual base classifier and the ensemble models created with varying classifier combination strategies.

4.2.1 *Performance of our proposed classifier ensemble model on Bengali dataset* Before presenting the performance of our proposed ensemble model on Bengali dataset, we need to discuss configuration of the base classifiers taking part in an ensemble. Since the base classifiers use either word $n$-grams or character $n$-grams as the features, the vocabulary size is the most important parameter for each base classifier. We have shown in this sub-section how vocabulary size affects the

[2] http://www.cs.waikato.ac.nz/ml/weka/

performance of a base classifier. As we discussed earlier, for each base classifier, the vocabulary is created by selecting the most frequent $k$ tokens per class and taking union of the features selected for the different classes. Figures 2–4 show the performances of the three different base classifiers on Bengali dataset when the value of $k$ is varied.

As we can see from Figure 2, the best performance of the first base classifier on Bengali dataset is obtained when the value of $k$ (# of word $n$-gram features) is set to 1,200.

As we can see from Figure 3 the best performance of the second base classifier on Bengali dataset is achieved when $k$ (# of character $n$-gram features) is set to any one of the following values: 12,000, 13,000, 14,000, 15,000 and 16,000, but we set the value of $k$ to 12,000 (the minimum possible value) to keep dimension as low as possible.

As we can see from Figure 4, the third base classifier performs the best when $k$ (# of unigram features) takes any one of the possible values: 1,200, 1,400, 1,600, 1,800, 2,000, 2,500 and 3,000, but we set the value of $k$ to 1,200 to keep the dimensions as low as possible.

After setting the parameters of the base classifiers to optimal possible values as mentioned earlier we apply non-trainable classifier combination rules to develop the heterogeneous ensemble model, which is evaluated on Bengali dataset, and the obtained results are shown in Table 2.

Table 2 shows that our proposed heterogeneous ensemble model with majority voting classifier combination rule performs the best on Bengali sentiment dataset among the model variants developed by changing the classifier combination rule. Table 3 shows that the proposed ensemble model performs better than each individual base classifier and hence, our proposed ensemble model is effective for sentiment analysis of Bengali tweets. We can also see from Table 3 that the second base classifier that uses Multinomial Naïve Bayes with sentiment lexicon and character $n$-gram features performs the best among the three base classifiers.

#### 4.2.2 *Performance of our proposed classifier ensemble model on Hindi dataset*

In this sub-section, we have shown how performance of the various base classifiers on Hindi dataset is affected when $k$ (number of features selected per class) is varied. As we can see from Figure 5, the best performance of the first base classifier on Hindi dataset is achieved when the value of $k$ is set to 1,800. Figure 6 shows that the best performance of the second base classifier on Hindi dataset is achieved when the value of $k$ is set to 7,000.

As we can see from Figure 7, the best performance of the third base model on Hindi dataset is achieved when the value of $k$ is set to 1,700.

It is evident from Figures 2–7 that the values of $k$ for which the different base classifiers perform the best vary from one language domain to another language domain. In our case, the three different base classifiers perform the best on Bengali dataset when the values of $k$ for the first, second and third base classifiers are set to 1200, 12000 and 1200, respectively. On the other hand, when they are applied on Hindi dataset, their best performances are achieved when the values of $k$ for the first, second and third base classifiers are set to 1800, 7000 and 1700, respectively. This phenomenon justifies the well-known ”no free lunch” theorem; that is, improvement of performance in problem-solving hinges on using some information to match procedures to problems [53].

However, in order to develop the heterogeneous classifier ensemble model for Hindi tweet sentiment classification, the base classifiers with the best parameter configurations are also constructed and then they are combined using various non-trainable classifier combination strategies as discussed earlier in this paper. The results obtained by the ensemble model for Hindi dataset are shown in Table 4. As we can see from Table 4, our proposed heterogeneous ensemble model with ”average of probabilities” performs the best among all model variants developed by changing the classifier combination rule. It is also evident from this table that performance of the model with ”majority voting” classifier combination rule is very close to that of the best model.

In Table 5, we have compared the performance of the best ensemble model for Hindi tweet sentiment analysis with the base classifier's individual performance. Table 5 shows that our proposed heterogeneous ensemble model with “average of probabilities” performs better than each individual classifier on Hindi dataset. Table 5 also shows that the ensemble model performs better on Hindi dataset than each individual base classifier. Among the three base classifiers, the base classifier that uses Multinomial Naïve Bayes with sentiment lexicon and character $n$-gram features performs the best on Hindi sentiment dataset also. Comparing performances of the base classifiers across the languages as shown in Tables 3 and 5, we can also observe that character $n$-gram features are more effective for sentiment analysis of tweets in both the Indian languages—Bengali and Hindi.

### 4.3 *Performance comparisons of our proposed system to some other existing systems for Bengali and Hindi sentiment analysis*

We have compared our proposed system to some other systems for Bengali sentiment analysis. To do so, we have implemented some existing methods published in the research papers. Our implementations of the existing methods have been evaluated on SAIL 2015 Bengali and Hindi datasets. For meaningful comparisons among the systems, 10-fold cross-validated results achieved by the systems are compared.

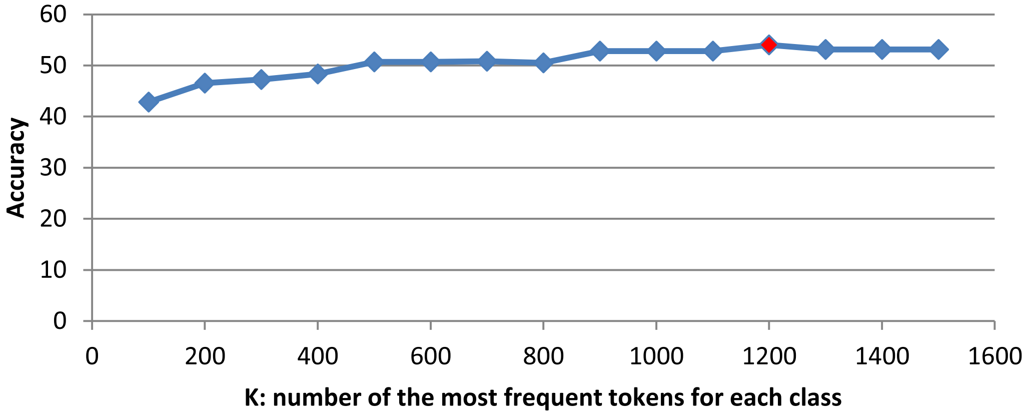


**Figure 2.** Performance of the first base classifier (Multinomial Naïve Bayes with sentiment lexicon and word *n*-gram features) vs. *k* on Bengali dataset

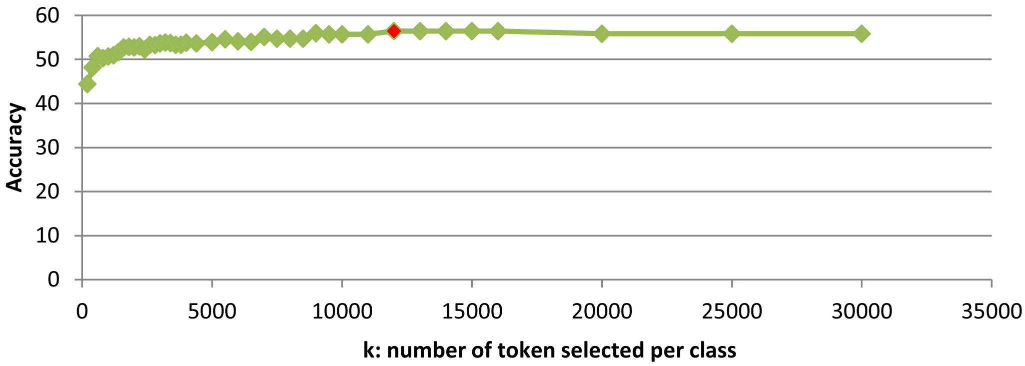


**Figure 3.** Performance of the second base classifier (Multinomial Naïve Bayes with sentiment lexicon and character *n*-gram features) vs. *k* on Bengali dataset

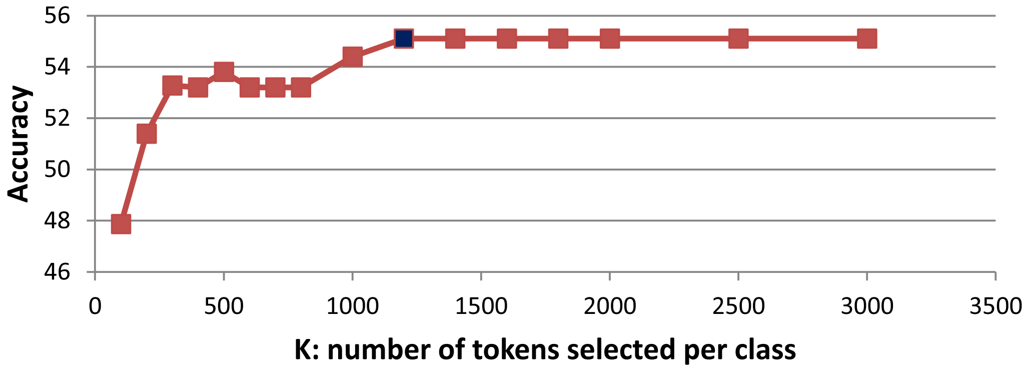


**Figure 4.** Performance of the third base classifier (linear SVM with sentiment lexicon and unigram features) vs. *k* on Bengali dataset

Two existing methods for Bengali sentiment polarity classification presented in [36] and [38] use Multinomial Naïve Bayes as the classifier. The performance comparison of Multinomial Naïve Bayes and SVM has also been presented in [36]. Though the models presented in [36, 38] are similar in some way to the base models used in developing our proposed ensemble model, our used base models differ in the feature sets. Initially, we have compared the performance of our proposed heterogeneous ensemble classification model to the three existing sentiment polarity classification models presented in [36, 38]. The performance comparison of these three existing models to our proposed ensemble model is shown in Tables 6 and 7. It is evident from Tables 6 and  7 that, for both the languages—

**Table 2.** Ten-fold sentiment classification performance of our proposed heterogeneous ensemble model on SAIL 2015 Bengali dataset

| Classifier combination strategies | Accuracy (%) |
|---|---|
| Majority voting | 57.53 |
| Average of probabilities | 57.13 |
| Maximum probability | 55.27 |

Bengali and Hindi, our proposed ensemble model performs better in terms of accuracy than the models presented in [36, 38].

For meaningful comparisons of our proposed ensemble model to existing methods for Bengali sentiment analysis task, we have conducted the two-tailed paired *t*-test to check whether the difference in mean accuracy achieved by our proposed ensemble model and the Bayesian model based on character *n*-gram presented in [38] is statistically

**Table 3.** Ten-fold sentiment classification performance of our proposed heterogeneous classifier ensemble model and the individual base classification models on SAIL 2015 Bengali dataset

| Systems | Accuracy (%) |
|---|---|
| Proposed heterogeneous classifier ensemble model with “”majority voting”” combination rule | 57.53 |
| Base mode 1 (Multinomial Naïve Bayes with sentiment lexicon and word *n*-gram features) | 54 |
| Base model 2 (Multinomial Naïve Bayes with sentiment lexicon and character *n*-gram features) | 56.4 |
| Base model 3 (SVM with sentiment lexicon and unigram features) | 55.1 |

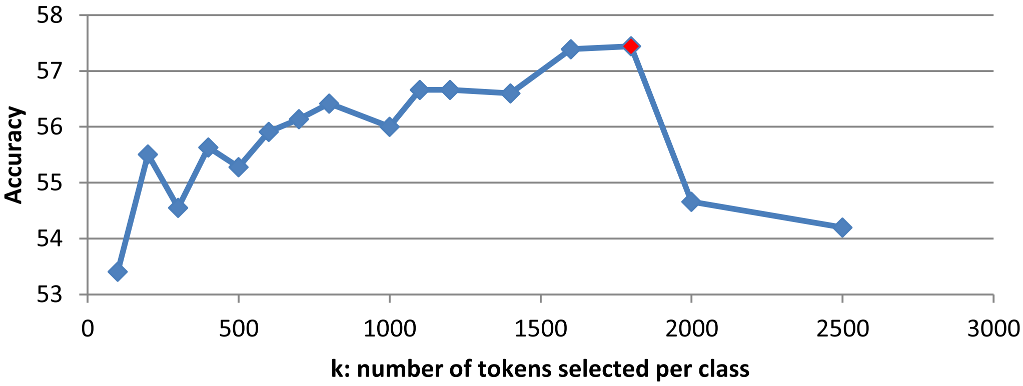


**Figure 5.** Performance of the first base classifier (Multinomial Naïve Bayes with sentiment lexicon and word *n*-gram features) vs. *k* on Hindi dataset

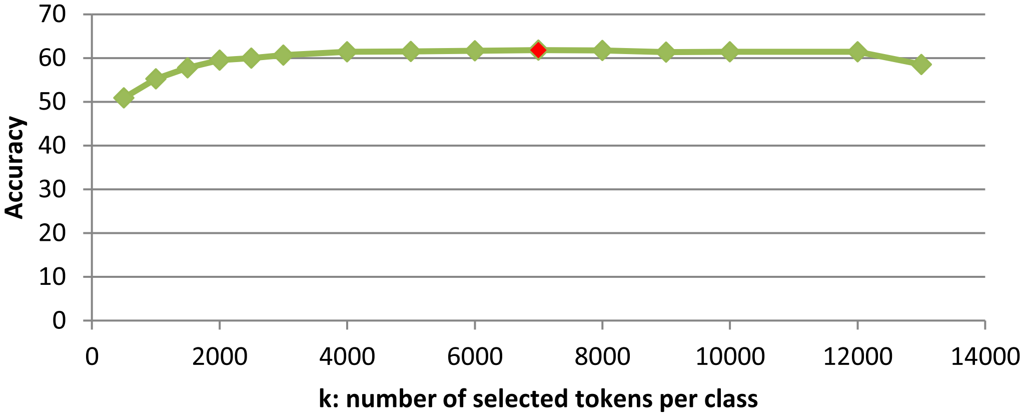


**Figure 6.** Performance of the second base classifier (Multinomial Naïve Bayes with sentiment lexicon and character *n*-gram features) vs. *k* on Hindi dataset

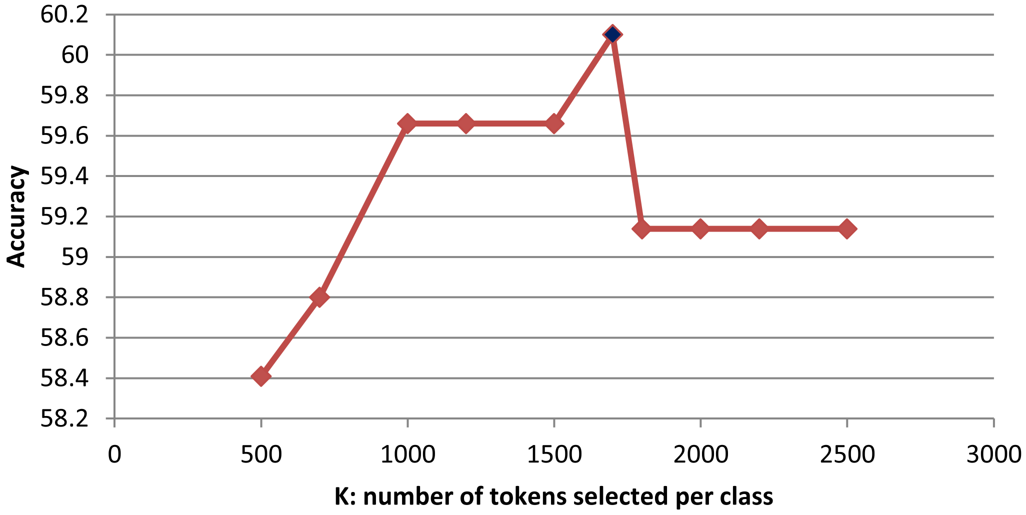


**Figure 7.** Performance of the third base classifier (linear SVM with sentiment lexicon and unigram features) vs. *k* on Hindi dataset

**Table 4.** Ten-fold sentiment classification performance of our proposed heterogeneous ensemble model on SAIL 2015 Hindi dataset

| Classifier combination strategies | Accuracy (%) |
|---|---|
| Average of probabilities | 62.63 |
| Majority voting | 62.16 |
| Maximum probability | 60.23 |

significant or not. The results of this significance test reveal that the difference is statistically significant ($P$ value is equal to 0.0085). Since our experiments using Hindi dataset show that the performance of SVM-based model presented in [36] is close to that of our proposed ensemble model, we also performed two-tail paired $t$-test to check whether the difference in mean accuracy for Hindi dataset achieved by the ensemble model and the SVM-based model is statistically significant or not. The results of this significance test also reveal that the difference is statistically significant ($P$ value is equal to 0.008).

Since of late the deep learning models are becoming popular for many text analysis tasks, we have also compared the sentiment classification performance of our proposed ensemble model to some deep learning models that are widely used for sentiment analysis for English language. Though many researchers have shown that deep-learning-based models are very effective for English sentiment analysis task, we find a limited number of research works that use deep learning models for Bengali and Hindi sentiment analysis tasks. One of the possible reasons is that

**Table 5.** Ten-fold sentiment classification performance of our proposed heterogeneous classifier ensemble model and the individual base classifiers on SAIL 2015 Hindi dataset

| Systems | Accuracy (%) |
|---|---|
| Proposed heterogeneous classifier ensemble model with “average of probabilities” combination rule | 62.63 |
| Base mode 1 (Multinomial Naïve Bayes with sentiment lexicon and word $n$-gram features) | 57.44 |
| Base model 2 (Multinomial Naïve Bayes with sentiment lexicon and character $n$-gram features) | 61.78 |
| Base model 3 (SVM with sentiment lexicon and unigram features) | 60.1 |

**Table 6.** Performance comparisons of our proposed model and the three existing machine learning models applied to Bengali tweet sentiment polarity classification

| Systems | Accuracy (%) |
|---|---|
| Proposed heterogeneous classifier ensemble model with ”majority voting”” combination rule | 57.53 |
| Model based on Multinomial Naïve Bayes in [38] | 55.2 |
| SVM-based model in [36] | 53.73 |
| Model based on Multinomial Naïve Bayes in [36] | 53.07 |

**Table 7.** Performance comparisons of our proposed model and the three existing machine learning models applied to Hindi tweet sentiment polarity classification

| Systems | Accuracy (%) |
|---|---|
| Proposed heterogeneous classifier ensemble model with ”average of probabilities” combination rule | 62.63 |
| SVM-based model presented in [36] | 59.65 |
| Model based on Multinomial Naïve Bayes in [38] | 53.75 |
| Model based on Multinomial Naïve Bayes in [36] | 43.75 |

deep learning algorithms are highly data hungry and the sentiment datasets for Indian languages are relatively small in size. However, the common deep learning algorithms that are widely used for sentiment analysis tasks are LSTM recurrent neural networks, BILSTM and CNN. Hence, we have implemented these three deep learning models for our sentiment analysis tasks and compared our proposed model to our developed deep-learning-based models. All the deep learning models have been implemented using Keras Python library. Each deep learning model has several tunable hyper-parameters, which need to be properly tuned to obtain the best possible results.

The model configurations for which our implemented deep-learning-based models give the best 10-fold cross-validated results are given here.

- LSTM-based model
  - embedding-size = 100
  - spatial dropout = 0.4
  - dropout = 0.2 and recurrent-dropout = 0.2
  - dimension of the output space = 100
  - loss function = ”categorical-crossentropy”
  - optimizer = ”Adam”
  - epochs = 25
  - batch-size =100
- Bidirectional LSTM model
  - embedding-size = 32
  - dropout = 0.2
  - dimension of the output space = 32
  - loss function = ”categorical-crossentropy”
  - optimizer = ”RmsProp”
  - epochs = 20
  - batch-size = 64
- CNN-based model
  - embedding size = 32
  - activation of CNN layer = ”relu”
  - number of filters = 32, kernel-size = 3, strides = 1
  - maximum pooling size = 3
  - size of the first dense layer = 32
  - activations of the dense layer = ”tanh
  - dropout = 0.2
  - loss function = “ categorical-crossentropy”
  - optimizer = ”Adam”
  - epochs = 20
  - batch-size = 64

The 10-fold cross-validated results of our developed deep learning models with the afore-mentioned configurations are shown in Tables 8 and 9. Table 8 shows performances comparisons of our proposed ensemble model with the deep-learning-based models for Bengali sentiment classification and Table 9 shows performances comparisons of our proposed ensemble model with the deep-learning-based models for Hindi sentiment classification.

As we can see from Tables 8 and 9, for both the languages, the BILSTM model performs the best among our implemented three deep learning models though its performance is slightly worse than that of our proposed ensemble model. To prove effectiveness of our proposed model, we conducted paired *t*-test to check whether difference in mean accuracy achieved by the proposed ensemble model and the BILSTM model is statistically significant or not. The results of the paired *t*-test show that the difference is statistically significant for Hindi language (*P* value is equal to 0.0173) as well as Bengali language (*P* value is equal to 0.0378).

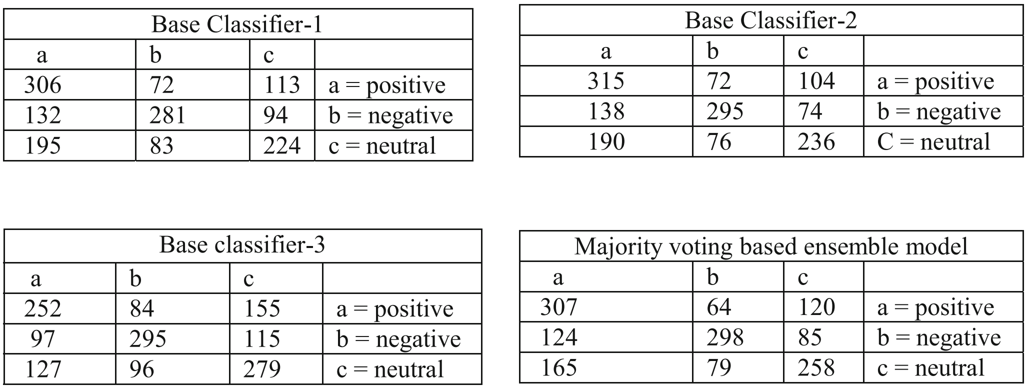
While implementing deep learning model for sentiment analysis task, the most common practice is to incorporate pre-trained embeddings in the deep learning model via transfer learning. One of the major downsides of using pre-trained embeddings is that the nature of data used for obtaining pre-trained embeddings is often very different from the data used to train the sentiment analysis model. To overcome this problem, the authors of some previous works [54, 55] suggest that deep learning model with transfer learning using polarized word embeddings is useful. Polarized embeddings are representations built on a corpus collected with a specific bias [54]. To obtain the polarized embeddings, they have acquired a large volume of comments and tweets specific to the domain and used traditional Word2Vec model [56] to obtain polarized embeddings. In our work, we follow this approach for generating polarized embeddings by collecting a large number of tweets since our focus is on developing sentiment analysis model for analysing tweet data. We have collected additional 3000 Bengali tweets and 5000 Hindi tweets. We have been able to collect relatively less number of tweets for Bengali language because we have seen that

**Table 8.** Performance comparisons of our proposed model with the deep-learning-based models for Bengali sentiment classification

| Systems | Accuracy (%) |
|---|---|
| Proposed heterogeneous classifier ensemble model with “”majority voting”” combination rule | 57.53 |
| BILSTM | 55.73 |
| LSTM-based model | 55.27 |
| CNN | 51.93 |
| BILSTM with polarity embedding | 54.87 |

**Table 9.** Performance comparisons of our proposed model with the deep-learning-based models for Hindi sentiment classification

| Systems | Accuracy (%) |
|---|---|
| Proposed heterogeneous classifier ensemble model with “”average of probabilities”” combination rule | 62.63 |
| BILSTM | 60.60 |
| LSTM-based model | 60.13 |
| CNN | 56.47 |
| BILSTM with polarity embedding | 59.67 |



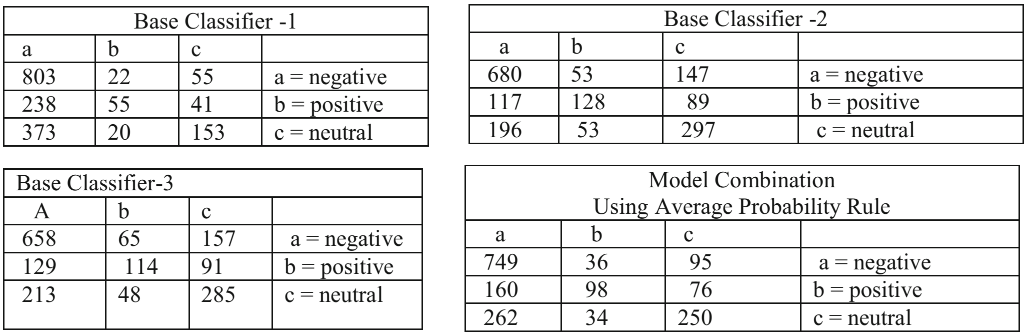
| Base Classifier-1 | | | |
|---|---|---|---|
| a | b | c | |
| 306 | 72 | 113 | a = positive |
| 132 | 281 | 94 | b = negative |
| 195 | 83 | 224 | c = neutral |

| Base Classifier-2 | | | |
|---|---|---|---|
| a | b | c | |
| 315 | 72 | 104 | a = positive |
| 138 | 295 | 74 | b = negative |
| 190 | 76 | 236 | C = neutral |

| Base classifier-3 | | | |
|---|---|---|---|
| a | b | c | |
| 252 | 84 | 155 | a = positive |
| 97 | 295 | 115 | b = negative |
| 127 | 96 | 279 | c = neutral |

| Majority voting based ensemble model | | | |
|---|---|---|---|
| a | b | c | |
| 307 | 64 | 120 | a = positive |
| 124 | 298 | 85 | b = negative |
| 165 | 79 | 258 | c = neutral |

**Figure 8.** Confusion matrices depicting performances of the three base classifiers and the ensemble model on Bengali sentiment dataset



| Base Classifier -1 | | | |
|---|---|---|---|
| a | b | c | |
| 803 | 22 | 55 | a = negative |
| 238 | 55 | 41 | b = positive |
| 373 | 20 | 153 | c = neutral |

| Base Classifier -2 | | | |
|---|---|---|---|
| a | b | c | |
| 680 | 53 | 147 | a = negative |
| 117 | 128 | 89 | b = positive |
| 196 | 53 | 297 | c = neutral |

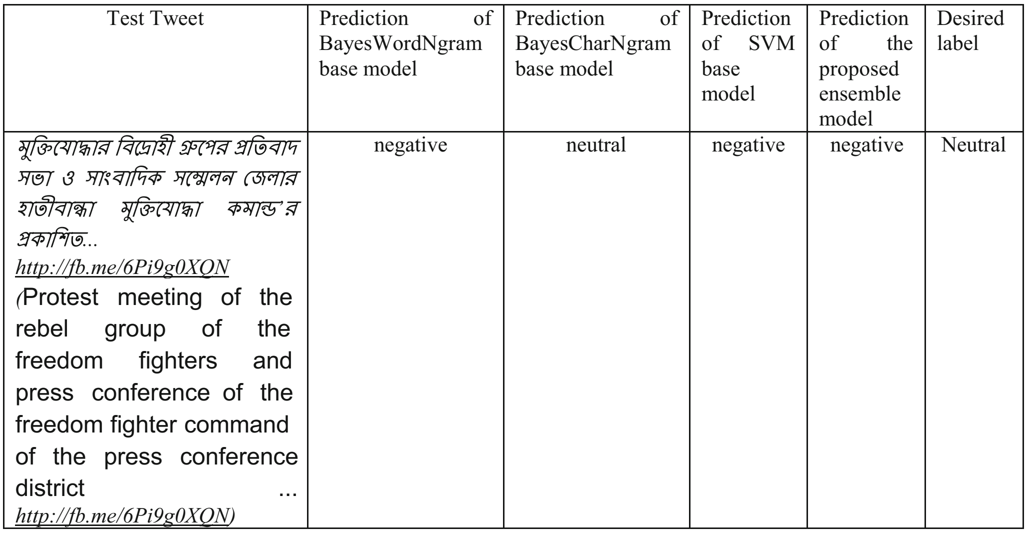
| Base Classifier-3 | | | |
|---|---|---|---|
| A | b | c | |
| 658 | 65 | 157 | a = negative |
| 129 | 114 | 91 | b = positive |
| 213 | 48 | 285 | c = neutral |

| Model Combination<br>Using Average Probability Rule | | | |
|---|---|---|---|
| a | b | c | |
| 749 | 36 | 95 | a = negative |
| 160 | 98 | 76 | b = positive |
| 262 | 34 | 250 | c = neutral |

**Figure 9.** Confusion matrices depicting performances of the three base classifiers and the ensemble model on Hindi sentiment dataset

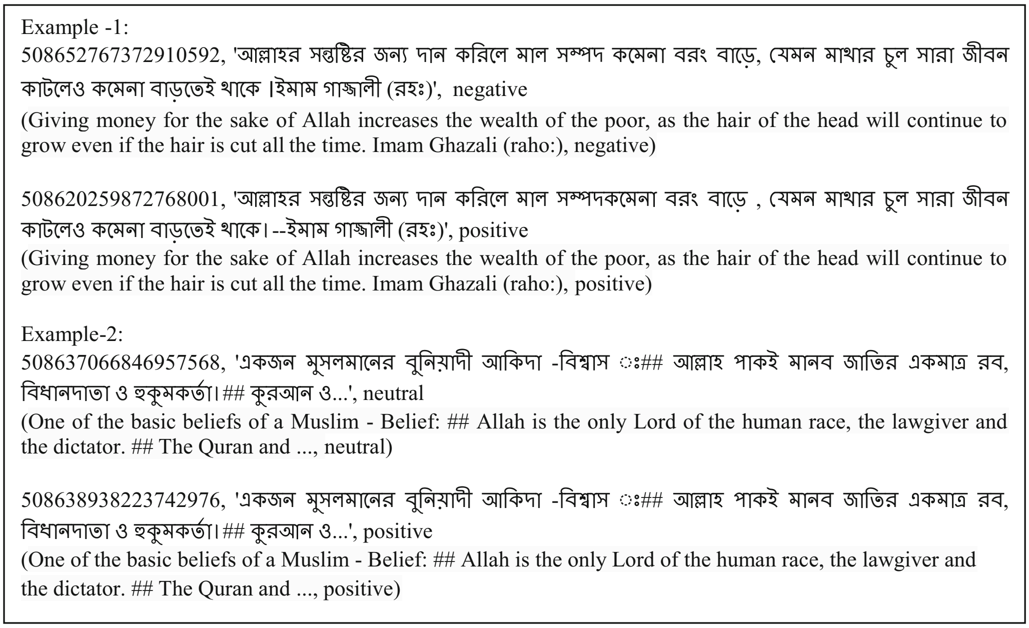
Bengali tweets are less frequent on Twitter. We have used Gensim[3]-based Word2Vec model for creating polarity embedding matrices for each language. After creating the polarized embeddings using the acquired data, we have developed the sentiment analysis model using BILSTM with polarity embedding. For this purpose, the embedding

[3] https://radimrehurek.com/gensim/models/word2vec.html



| Test Tweet | Prediction of BayesWordNgram base model | Prediction of BayesCharNgram base model | Prediction of SVM base model | Prediction of the proposed ensemble model | Desired label |
|---|---|---|---|---|---|
| মুক্তিযোদ্ধার বিদ্রোহী গ্রুপের প্রতিবাদ সভা ও সাংবাদিক সম্মেলন জেলার হাতীবান্ধা মুক্তিযোদ্ধা কমান্ড'র প্রকাশিত... http://fb.me/6Pi9g0XQN<br>(Protest meeting of the rebel group of the freedom fighters and press conference of the freedom fighter command of the press conference district ... http://fb.me/6Pi9g0XQN) | negative | neutral | negative | negative | Neutral |

**Figure 10.** An example of a tweet misclassified by our proposed ensemble model



Example -1:
508652767372910592, 'আল্লাহর সন্তুষ্টির জন্য দান করিলে মাল সম্পদ কমেনা বরং বাড়ে, যেমন মাথার চুল সারা জীবন কাটলেও কমেনা বাড়তেই থাকে ।ইমাম গাজ্জালী (রহঃ)', negative
(Giving money for the sake of Allah increases the wealth of the poor, as the hair of the head will continue to grow even if the hair is cut all the time. Imam Ghazali (raho:), negative)

508620259872768001, 'আল্লাহর সন্তুষ্টির জন্য দান করিলে মাল সম্পদকমেনা বরং বাড়ে , যেমন মাথার চুল সারা জীবন কাটলেও কমেনা বাড়তেই থাকে। --ইমাম গাজ্জালী (রহঃ)', positive
(Giving money for the sake of Allah increases the wealth of the poor, as the hair of the head will continue to grow even if the hair is cut all the time. Imam Ghazali (raho:), positive)

Example-2:
508637066846957568, 'একজন মুসলমানের বুনিয়াদী আকিদা -বিশ্বাস ◌ঃ## আল্লাহ পাকই মানব জাতির একমাত্র রব, বিধানদাতা ও হুকুমকর্তা। ## কুরআন ও...', neutral
(One of the basic beliefs of a Muslim - Belief: ## Allah is the only Lord of the human race, the lawgiver and the dictator. ## The Quran and ..., neutral)

508638938223742976, 'একজন মুসলমানের বুনিয়াদী আকিদা -বিশ্বাস ◌ঃ## আল্লাহ পাকই মানব জাতির একমাত্র রব, বিধানদাতা ও হুকুমকর্তা। ## কুরআন ও...', positive
(One of the basic beliefs of a Muslim - Belief: ## Allah is the only Lord of the human race, the lawgiver and the dictator. ## The Quran and ..., positive)

**Figure 11.** Some annotation errors

matrix is transferred to the BILSTM layer via an embedding layer and it is further fine-tuned with the help of training dataset for the respective language by setting "trainable = true" in the Keras embedding layer. However, the experimental results show that transfer learning using polarity embedding does not improve the sentiment classification performance on our datasets; rather, the performance slightly drops for both the languages—Bengali and Hindi. The obtained results for our implemented BILSTM model with polarity embedding on Bengali and Hindi datasets are also shown in the last rows of Tables 8 and 9, respectively.

### 4.4 *Error analysis and discussion*

In this section, we have analysed why our proposed ensemble performs well for Bengali and Hindi sentiment analysis tasks and what are the possible reasons of misclassification error made by our proposed ensemble model.

To analyse why our proposed ensemble model performs well for Bengali and Hindi sentiment analysis tasks, we analysed the confusion matrices produced by individual base classifier and the ensemble model for both the languages. Figure 8 shows confusion matrices for Bengali language dataset. Each confusion matrix shown is the sum of 10 confusion matrices produced by the concerned model when it is run 10 times for the 10 different folds.

From the confusion matrices shown in Figure 8, we can observe that base classifier-1 and base classifier-2 are better in classifying the positive samples as positive, base classifier-2 and base classifier-3 are better in classifying negative samples as negative and base classifier-2 and base classifier-3 are better in classifying neutral samples as neutral. Hence, it is evident from these confusion matrices that the performance of the ensemble model would be better than that of any individual base classifier.

We have also shown in Figure 9 the confusion matrices produced by the individual base classifiers and the ensemble model for Hindi sentiment dataset. From Figure 9, we can see that base classifier-1 and base classifier-2 are better in classifying positive Hindi tweet as positive, base classifier-2 and base classifier-3 are better in classifying negative Hindi tweet as negative and base classifier-2 and base classifier-3 are better in classifying neutral Hindi tweet as neutral. By analysing the performances of the base classifiers, we can observe that the model combining all these base classifiers improves Hindi sentiment classification performance. We can also see from Figure 9 that combining base classifiers using ”average of probability” rule has led to better performance than that of any individual base classifier.

By analysing the confusion matrices produced by our developed models, we can conclude that each model has a general tendency of misclassifying the tweets belonging to neutral class. One of the possible reasons is that neutrality lies between the boundary of positivity and negativity and hence most annotation errors may occur while annotating the tweets as neutral class.

Other than the confusion matrices, we have also inspected some of the tweets misclassified by our proposed ensemble model. For example, the test example shown in Figure 10 is misclassified by the ensemble model as “negative” though the desired label of the tweet is “neutral”. It is clear from Figure 10 that the majority-voting-based ensemble model says “negative” because two out of three base models say “negative”.

To find the reasons of such errors we analyse the selected features and polarity of the individual tweet words, which are taken into account by the individual classifiers and we have the following observations. For the test example shown in Figure 10, BayesWordNgram- and SVM-based models remove some tokens during feature selection (for example ). Out of the selected tokens, two tokens are of negative polarity (for example ) and the other tokens are almost equally distributed among the three groups of tweets belonging to the three different sentiment classes. Hence, due to presence of two negative words in the tweet, both the base models classify the tweet as “negative”. However, we observe that BayesCharNgram model correctly predicts the class of the example as neutral. This is because data sparseness problem is resolved when the character *n*-gram features are taken into consideration. However, the test example is finally classified as “negative” by the ensemble model since two base models agree on class label ”negative”. We also find that the reason of classification error for Hindi dataset is the same as described earlier.

We have identified another possible reason for classification error. We have found that the SAIL 2015 dataset contains some annotation errors. Some examples of annotation errors are listed in Figure 11. As we can see from Figure 11 the contents of the tweets with IDs 508652767372910592 and 508620259872768001 are the same, but they have been annotated with the different sentiment class labels. This is true for another pair of tweets shown in the figure with IDs 508637066846957568 and 508638938223742976.

## 5. Conclusions and future works

In this paper, we have described a heterogeneous ensemble model for Bengali and Hindi tweet sentiment classification. Multinomial Naïve Bayes with word *n*-gram features, Multinomial Naïve Bayes with character *n*-gram features and SVM with unigram features have been combined in an ensemble using various classifier combination rules. Sentiment lexicon has also been incorporated in our proposed model. We observe that, among the classifier combination rules, “”majority voting”” rule is more effective for Bengali tweet sentiment analysis whereas “”average of probabilities”” rule is effective for Hindi tweet sentiment analysis.

We also observe that Multinomial Naïve Bayes with sentiment lexicon and character *n*-gram features performs the best among the three base classifiers we considered.

We have compared our proposed ensemble model to some deep learning models implemented by us and observed that the deep learning models have not performed well on our Bengali and Hindi datasets. Our proposed ensemble model has performed the best on SAIL 2015 datasets. One of the reasons of poor performance of the deep learning models may be the lack of sufficient training data. However, since SAIL 2015 datasets are the only benchmark datasets for Bengali and Hindi datasets, we think that the performance of the proposed ensemble model on SAIL 2015 datasets is encouraging and our experimental results show that the ensemble of classifiers with diverse set of features are effective for Indian language sentiment analysis tasks.

One of the problems that may have affected the system performance is that SAIL 2015 dataset contains some annotation errors, that is, some tweets have been wrongly labelled by the human annotators. However, for meaningful comparisons of our proposed system with some existing systems, we did not correct those annotation errors manually. We hope that more training data and proper annotation will help in future to develop a more accurate sentiment polarity detection system for Bengali and Hindi tweets. We also hope that our proposed system can easily be extended to other Indian languages like Tamil, Telugu, Marathi, etc. with minor modifications.

Sometimes Twitter posts becomes ironic and failing to detect irony can lead to low performance for sentiment analysis system, since the presence of irony often causes polarity reversal [57]. Hence, our future plan is to investigate how irony detection can help in analysing sentiments of Bengali and Hindi tweets. Like irony detection, fake review or opinion detection [58, 59] is also necessary for making the sentiment analysis more useful in practice. Hence, in future, we will investigate how to integrate fake opinion filtering and irony detection with our proposed sentiment analysis method for developing a more accurate and practical sentiment analysis system.

## Acknowledgements

This research work is supported by the project titled "Indian Social Media Sensor: An Indian Social Media Text Mining System for Topic Detection, Topic Sentiment Analysis and Opinion Summarization" funded by the Department of Science and Technology, Government of India, under the SERB scheme.

## References

[1] Bowker J 2003 *The concise Oxford dictionary of world religions*. Oxford University Press, Oxford

[2] Zhao J, Liu K, Wang G 2008 Adding redundant features for CRFs-based sentence sentiment classification. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 117–126

[3] Joachims T 1998 *Making large scale SVM learning practical*. Technical Report

[4] Pang B, Lee L, Vaithyanathan S 2002 Thumbs up? Sentiment classification using machine learning techniques. In: *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, vol. 10, pp. 79–86

[5] Dave K, Lawrence S, Pennock D M 2003 Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: *Proceedings of the 12th International Conference on World Wide Web*, pp. 519–528

[6] Mullen T, Collier N 2004 Sentiment analysis using support vector machines with diverse information sources. In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 412–418

[7] Pang B, Lee L 2008 Opinion mining and sentiment analysis. In: *Foundations and Trends in Information Retrieval*. Now Publishers Inc., vol. 2(1–2), pp. 1–135

[8] Goldberg A B, Zhu X 2006 Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In: *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, Association for Computational Linguistics, pp. 45–52

[9] Miao Q, Li Q, Zeng D 2010 Fine-grained opinion mining by integrating multiple review sources. *Journal of the American Society for Information Science and Technology* 61(11): 2288–2299

[10] Riloff E, Wiebe J 2003 Learning extraction patterns for subjective expressions. In: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pp. 105–112

[11] Prabowo R, Thelwall M 2009 Sentiment analysis: a combined approach. *Journal of Informetrics* 3(2): 143–157

[12] Narayanan R, Liu B, Choudhary A 2009 Sentiment analysis of conditional sentences. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing:*, Association for Computational Linguistics, vol. 1, pp. 180–189

[13] Wiegand M, Balahur A, Roth B, Klakow D, Montoyo A 2010 A survey on the role of negation in sentiment analysis. In: *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, pages 60–68

[14] Ku L-W, Liang Y-T, Chen H-H 2006 Opinion extraction, summarization and tracking in news and blog corpora. In: *Proceedings of AAAI*, pp. 100–107

[15] Kim J, Chern G, Feng D, Shaw E, Hovy E 2006 Mining and assessing discussions on the web through speech act analysis. In: *Proceedings of the Workshop on Web Content Mining with Human Language Technologies at the 5th International Semantic Web Conference*, pp. 5–9

[16] Pang B, Lee L 2004 A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, p. 271

[17] Zhu F, Zhang X 2010 Impact of online consumer reviews on sales: the moderating role of product and consumer characteristics. *Journal of Marketing* 74(2): 133–148

[18] Melville P, Gryc W, Lawrence R D, 2009 Sentiment analysis of blogs by combining lexical knowledge with text classification. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1275–1284

[19] Ramakrishnan G, Jadhav A, Joshi A, Chakrabarti S, Bhattacharyya P 2003 Question answering via Bayesian inference on lexical relations. In: *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*, Association for Computational Linguistics, vol. 12, pp. 1–10

[20] Jiao J, Zhou Y 2011 Sentiment polarity analysis based multi-dictionary. *Physics Procedia* 22: 590–596

[21] Macdonald C, Ounis I 2006 *The TREC Blogs06 collection: creating and analysing a blog test collection*. Tech Report TR-2006-224, Department of Computer Science, University of Glasgow, vol. 1, pp. 3–1

[22] Hatzivassiloglou V, McKeown K R 1997 Predicting the semantic orientation of adjectives. In: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European*

*Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 174–181

[23] Wiebe J M 2000 Learning subjective adjectives from corpora. In: *Proceedings of AAAI/IAAI*, pp. 735–740

[24] Yu H, Hatzivassiloglou V 2003 Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 129–136

[25] Esuli A, Sebastiani F 2006 Sentiwordnet: a publicly available lexical resource for opinion mining. In *Proceedings of LREC*, vol. 6, pp. 417–422

[26] Fellbaum C 1999 In: *WordNet*. Blackwell Publishing Ltd.

[27] Chen C C, Tseng Y-D 2011 Quality evaluation of product reviews using an information quality framework. *Decision Support Systems* 50(4): 755–768

[28] Kang H, Yoo S J, Han D 2012 Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems with Applications* 39(5): 6000–6010

[29] Sarkar K, Chakraborty S 2015 A sentiment analysis system for Indian language tweets. In: *Proceedings of the International Conference on Mining Intelligence and Knowledge Exploration*. Springer, pp. 694–702

[30] Onan A, Korukoglu S, Bulut H 2016 A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. *Expert Systems with Applications* 62: 1–16

[31] da Silva N F F, Hruschka E R, Hruschka E R 2014 Tweet sentiment analysis with classifier ensembles. *Decision Support Systems* 66: 170–179

[32] Rodríguez-Penagos C, Batalla J A, García-Narbona J C-F D, Grivolla J, Lambert P, Sauri R 2013 FBM: combining lexicon-based ML and heuristics for social media polarities. In: *Second Joint Conference on Lexical and Computational Semantics (*SEM). Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, vol. 2, pp. 483–489

[33] Hassan A, Abbasi A, Zeng D 2013 Twitter sentiment analysis: a bootstrap ensemble framework. In: *Proceedings of the 2013 International Conference on Social Computing*, IEEE, pp. 357–364

[34] Ankit S N 2018 An ensemble classification system for twitter sentiment analysis. *Procedia Computer Science* 132: 937–946

[35] Patra B G, Das D, Das A, Prasath R 2015 Shared task on sentiment analysis in Indian languages (SAIL) tweets—an overview. In: *Proceedings of the International Conference on Mining Intelligence and Knowledge Exploration*. Springer, pp. 650–655

[36] Sarkar K, Bhowmick M 2017 Sentiment polarity detection in bengali tweets using multinomial Naïve Bayes and support vector machines. In: *Proceedings of the 2017 IEEE Calcutta Conference (CALCON)*, IEEE, pp. 31–36

[37] Sarkar K 2019 Sentiment polarity detection in Bengali tweets using deep convolutional neural networks. *Journal of Intelligent Systems* 28(3): 377–386

[38] Sarkar K 2018 Using character *N*-gram features and Multinomial Naïve Bayes for sentiment polarity detection in Bengali tweets. In: *Proceedings of the 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)* , pp. 1–4

[39] Sarkar K 2019 Sentiment polarity detection in Bengali tweets using LSTM recurrent neural networks. In: *Proceedings of the 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, pp. 1–6

[40] Sharma Y, Mangat V, Kaur M 2015 A practical approach to sentiment analysis of Hindi tweets. In: *Proceedings of the 2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, pp. 677–680

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

[42] Mittal N, Agarwal B, Chouhan G, Bania N, Pareek P 2013 Sentiment analysis of Hindi reviews based on negation and discourse relation. In: *Proceedings of the 11th Workshop on Asian Language Resources*

[43] Ouyang X, Zhou P, Li C H, Liu L 2015 Sentiment analysis using convolutional neural network. In: *Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 2359–2364

[44] Wang X, Jiang W, Luo Z 2016 Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts. In: *Proceedings of COLING*

[45] Rani S, Kumar P 2019 Deep learning based sentiment analysis using convolution neural network. *Arabian Journal for Science and Engineering* 44: 3305–3314

[46] Singh J, Singh G, Singh R, Singh P 2018 Morphological evaluation and sentiment analysis of Punjabi text using deep learning classification. *Journal of King Saud University – Computer and Information Sciences*

[47] Akhtar M S, Kumar A, Ekbal A, Bhattacharyya P 2016 A hybrid deep learning architecture for sentiment analysis. In: *Proceedings of COLING*, pp. 482–493

[48] Kittler J, Hatef M, Duin R P W 1996 Combining classifiers. In: *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 2, pp. 897–901

[49] Das A, Bandyopadhyay S 2010 SentiWordNet for Indian Languages. In: *Proceedings of COLING*, pp. 56–63

[50] Vapnik V 1982 Estimation of dependences based on empirical data. In: *Springer Series in Statistics*. Springer-Verlag, vol. 40

[51] Yang Y, Liu X 1999 A re-examination of text categorization methods. In: *Proceedings of SIGIR '99*

[52] Platt J C 1999 Fast training of support vector machines using sequential minimal optimization, advances in kernel methods. In: *Advances in kernel methods: support vector learning*

[53] Wolpert D H, Macready W G 1997 No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1: 67–82

[54] Merenda F, Zaghi C, Caselli T, Nissim M 2018 Source-driven representations for hate speech detection. In: *Proceedings of the 5th Italian Conference on Computational Linguistics*, Turin, Italy

[55] Graumans L, David R, Caselli T 2019 Twitter-based polarised embeddings for abusive language detection. In: *Proceedings of the 2019 8th International Conference on*

*Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pp. 1–7

[56] Mikolov T, Chen K, Corrado G S, Dean J 2013 Efficient estimation of word representations in vector space. In: *Proceedings of CoRR*, abs/1301.3781

[57] Zhang S, Zhang X, Chan J, Rosso P 2019 Irony detection via sentiment-based transfer learning. *Information Processing and Management* 56: 1633–1644

[58] Cagnina L C, Rosso P 2017 Detecting deceptive opinions: intra and cross-domain classification using an efficient representation. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 25(2): 175–189

[59] Rosso P, Cagnina L C 2017 Deception detection and opinion spam. In: Cambria E, Das D, Bandyopadhyay S and Feraco S (Eds.) *A Practical Guide to Sentiment Analysis. Socio-Affective Computing*, vol. 5. Springer-Verlag, pp. 155–171