[Turkish Journal of Electrical Engineering and Computer Sciences](https://journals.tubitak.gov.tr/elektrik)

Volume 25 | Number 3 Article 38

1-1-2017

# Intellimote: a hybrid classifier for classifying learners' emotion in a distributed e-learning environment

LOPA MANDAL

ROHAN DAS

SAMAR BHATTACHARYA

PRAMATHA NATH BASU

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 2084 – 2095

doi:10.3906/elk-1510-120

# Intellimote: a hybrid classifier for classifying learners' emotion in a distributed e-learning environment

**Lopa MANDAL**[1,*], **Rohan DAS**[1], **Samar BHATTACHARYA**[2], **Pramatha Nath BASU**[3]
[1]Department of Information Technology, Institute of Engineering & Management, Kolkata, India
[2]Department of Electrical Engineering, Faculty of Engineering and Technology, Jadavpur University, Kolkata, India
[3]School of Education Technology, Faculty of Engineering and Technology, Jadavpur University, Kolkata, India

**Received:** 16.10.2015 • **Accepted/Published Online:** 18.08.2016 • **Final Version:** 29.05.2017

**Abstract:** A huge collection of textual, graphical, audio, and video contents are readily available on the Internet to be used for the purpose of learning. Sentimental feedbacks of learners posted at the end of many of these contents may be considered as genuine reactions of the learners who have gone through the contents. Such learners' sentiments are important inputs for judging the acceptability of a learning material. Analyzing such feedbacks using sentiment analysis techniques can identify the best reusable learning contents that may be used for developing new courseware. This can significantly reduce the time and effort of authoring, which is otherwise a difficult, time-consuming, and costly affair. This methodology can also be used for continuous assessment on the learning materials released for use. This paper presents a machine learning-based approach for emotion analysis in e-learning materials. It describes the design and experimental use of a sentiment analysis classifier that uses classifier combination rules to combine polarity scoring and a support vector machine (SVM). The present approach also gives an opportunity for users to train a lexicon on a very specialized set of data, pertaining to the domain of usage. This helps either to enhance polarity scores of certain words that appear more frequently or to add words that are completely missing from the lexicon, and contributes a great deal in determining the polarity scores.

**Key words:** Sentiment analysis, opinion mining, hybrid classifier, polarity scoring, support vector machines, feedback analysis

## 1. Introduction

In a physical classroom environment, teachers and instructors can observe students' body language and facial expressions to understand their feedback about a lecture session through conversations and interactions. Some questionnaires may also be given to them to know their feedback about the content delivered. However in an e-learning environment, students need to establish an online presence or fingerprint that can be understood and accessed by instructors. Sentiment analysis or the computational study of opinions, sentiments, and emotions in a text can be a very effective solution in a large-scale, diverse, distributed e-learning environment. The expressive power of this type of feedback is much more in comparison to other techniques.

Authoring effective contents for developing courseware is a difficult, time-consuming, and costly affair. Continuous quality assessment of the existing contents, already released for use, is another important requirement. In this context, the present work explores sentiment analysis-based methodologies for feedback analysis

*Correspondence: mandal.lopa@gmail.com

to identify useful tested learning content from a huge collection of free online material, for the purpose of reusing it in authoring courseware. This is expected to reduce the time, cost, and effort of the courseware development process. It also explores methodology for continuous assessments of developed materials, already released for use, which may need modifications depending on the learners' feedback, over a period of time.

## 2. Previous work

Sentiment analysis or the computational study of opinions, sentiments, and emotions in a text is already in use [1]. There are several approaches of sentiment analysis and a very broad overview describing the main techniques and approaches was presented by Pang and Lee [2]. Initially most of the work was focused on determining the semantic orientation of documents with an attempt to learn a positive/negative classifier at a document level. However, a lot of work was later derived from Turney's introduction of the results of review classification by considering the algebraic sum of the orientation of terms with respect to the orientation of documents [3]. Wilson et al. [4] developed a technique focused on specific tasks such as finding the sentiment of words.

Baroni and Vegnaduzzo [5] developed a technique that ranked a large list of adjectives according to a subjectivity score by employing a small set of manually selected subjective adjectives and computed the mutual information of pairs of adjectives using frequency and co-occurrence frequency counts on the web. Turney and Littman [6] proposed an approach to measure the semantic orientation of a given word based on the strength of its association with a set of context insensitive positive words minus the strength of its association with a set of negative words. Gamon and Aue [7] used this approach to build sentiment lexicons and assigned a sentiment polarity score to each word. Pang et al. [8] proposed three machine learning approaches: naïve Bayes, maximum entropy, and SVMs, to label the polarity of a movie review dataset. Combination of rule-based classification using supervised and machine learning, proposed by Prabowo and Thelwall [9], is a promising approach. Shein [10] introduced an SVM-based classification technique of documents. A different approach to social media content was to infer the sentiment orientations of the content and estimate the sentiment orientations of a collection of documents as a text classification problem [11]. Sentiment-related information can also be encoded within the actual words of the sentence through changes in attitudinal shades of word meaning using suffixes [12]. Wang et al. proposed a dictionary approach wherein the authors attempted to construct an emotion dictionary through a clustering model built with extracted emoticons as seeds from a large-scale corpus [13]. Following this, extracted new words are chosen as new seeds and iterated until convergence. The constructed emotion lexicon is combined with an SVM to predict the emotion polarity of a document. Some of the prebuilt dictionaries in use are: General Inquire [14], the Dictionary of Affect in Language [15], the WordNet-Affect [16], and SentiWordNet [12]. Poria et al. built a framework by means of fuzzy c-means clustering and SVM classification that takes into account a number of similarity measures, including point-wise mutual information and emotional affinity [17]. Unlike traditional approaches, which are mostly based on statistical methods, Li and Xu tried to infer and extract the reasons of emotions by importing knowledge and theories from other fields such as sociology [18]. Based on the theory that a triggering-cause event is an integral part of emotion, the technique of emotion-cause extraction is used as a crucial step to improve the quality of selected features. Bravo-Marquex et al. [19] proposed an approach that identified scenarios in which some lexical resources are more useful than others. Furthermore, they proposed a novel approach for sentiment classification based on meta-level features. This supervised approach boosted existing sentiment classification of subjectivity and polarity detection on social data mined from Twitter. Jian et al. proposed a new sampling technique based on the support vectors (SVs) and the nonsupport vectors (NSVs) for the purpose of classification [20]. Peker proposed k-medoids clustering-based attribute weighting (kmAW) as a data preprocessing method where SVM was used in the classification phase. The performance of the system was evaluated on classification accuracy, specificity, sensitivity analysis, f-measure, kappa statistics value, and ROC and was found to give better results in comparison to the other existing systems [21].

## 3. Materials and methods

The present work presents a sentiment analysis based approach for classifying learners' sentiment towards e-learning materials. The contents with the learners' comments (and/or suggestions/feedback) are collected and comments are passed through the proposed sentiment analysis engine. The sentiment analysis engine analyses the comments to classify them into positive, negative, or neutral. The results of the sentiment analysis are used to classify the learning objects based on subjectivity. The selected contents are stored in the content repository for the purpose of reuse. The overall process is shown in Figure 1.

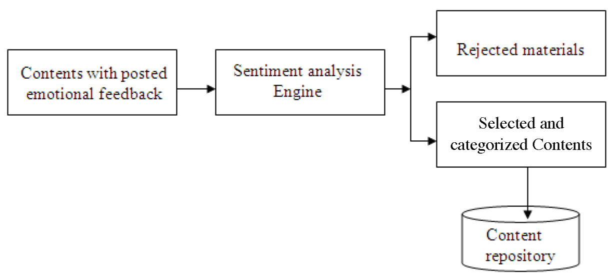


**Figure 1.** Block diagram of the methodology to select the content for reuse.

Continuous assessment of acceptability for newly developed materials is as important as that for the existing materials. For that purpose the learners need to be allowed to provide feedback on the learning materials, which can be captured and analyzed using the proposed sentiment analysis engine in the same way as earlier. Based on the analysis reports the content owners/domain experts can take necessary steps to modify and/or replace content as necessary. The overall flow of the systems is shown in Figure 2.

### 3.1. Sentiment analysis engine

The sentiment analysis engine is implemented using principles of natural language processing using a combination of polarity scoring and NuSVC implementation of the SVM. The SVM model uses the radial basis function (RBF) kernel. Regularization parameter C and kernel parameter $\gamma$ were determined by the hit and trial method. Generally the learners' feedback or comments in the English language are passed as input to the engine, wherein sentiment analysis takes place.

#### 3.1.1. Text preprocessing

Input of learners' feedback goes through several layers of preprocessing, so as to present the most relevant words to the analyzer for sentiment processing. Basically the role of the text preprocessor is to create a feature vector for further processing from the text. The following levels of preprocessing have been applied to the text:

- Tokenization (LEX): The text is tokenized into a list of words with all spaces removed. It is to be noted

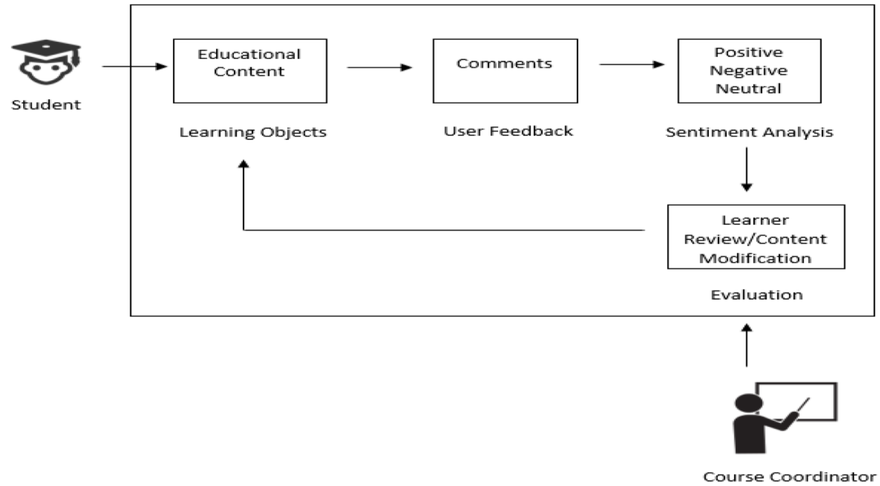


**Figure 2.** Use of feedback analysis in e-learning system.

that for the moment punctuation is retained, although punctuation does not play any role in the present feature vector.

- Removal of repeated characters (RRC): In emotion expression, people often do not remain strictly grammatical. They will write things such as “I looooooove it” to emphasize the word “love”. However, computers will not know that ”looooooove” is a variation of ”love”. This level presents a method to remove these annoying repeating characters in order to end up with a proper English word.

  The method uses regular expression to match and define a replacement string with back references.

- Negation (NEG): Append a _NEG suffix to every word appearing between a negation and a clause-level punctuation mark.

The sentiment tokenization strategy used here makes this straightforward, since it isolates the clausal punctuation from the word-internal punctuation.

The algorithm captures simple negations:

Example Text: No one enjoys it.

The tokens accounting for negation are: ‘no’, ‘one_NEG’, ‘enjoys_NEG’, ‘ it_NEG ’

It also handles long-distance effects: “I don’t think I will enjoy it, it might be too spicy”.

The tokens accounting for negation are: ‘i’, ‘don’t’, ‘think_NEG’, ‘i_NEG’, ‘will_NEG’, ‘enjoy_NEG’, ‘it_NEG’, ‘,’ , ‘it’, ‘might’, ‘be’, ‘too’, ‘spicy’.

- Removal of Stop words (STP): Using Treebank’s corpus of stop-words insignificant words in the given text are searched and filtered out. Words that generally do not contribute to the overall sentiment of the text are eliminated (e.g., articles, be verbs etc.). Stop-words are words that are often useless in NLP, in that they do not convey much meaning, such as the word ‘the’.

- Parts of Speech Tagging (POS): A POS tagger is trained using Treebank’s corpus of tagged sentences (3000 sentences). This tagger is used to tag the list of tokens into various parts of speech. All tokens/words that are not adjectives, adverbs, verbs, or untagged are removed from the feature vector. All existing punctuation is removed at this stage as well. To recollect, punctuation was initially retained to assist in negation tagging.

### 3.2. Polarity lexicon

SentiWordNet [12] attaches positive and negative real-valued sentiment scores to WordNet synsets [16]. The polarity lexicon is used for polarity scoring of user feedback in the first step of sentiment analysis to give an overall sentiment score to the user's feedback or comments. However, one may choose to use a training dataset to create a completely new lexicon. The present approach uses SentiWordNet as the base lexicon and then uses a training dataset to either enhance polarity scores of certain words that appear more frequently or to add words that are completely missing from the lexicon. This contributes a great deal in determining the polarity scores. Ideally the base SentiWordNet lexicon is sufficient to make most polarity decisions; however, if a user chooses to train a lexicon on a very specialized set of data, pertaining to a specific domain (e.g., educational data), then having the option to either enhance the existing lexicon or to create a new lexicon makes a great deal of sense. Either of these options helps in achieving better accuracy depending upon the domain of the data. The columns of the SentiWordNet lexicon that are used in the present work are parts of speech, the word itself, and the corresponding negative and positive polarity score.

### 3.3. Architecture of the classifier

A combination of polarity scoring and SVM classifier has been used in the proposed sentiment analysis engine. It is essentially a two-stage classifier as depicted in Figure 3. It is to be noted that the implementation of polarity scoring follows an overall sentiment score for any given piece of text/feedback, i.e. each text postprocessing has an overall positive or negative score as will be evident from the algorithm discussed below. A given feedback/text is rejected or assumed to be unclassified by the polarity scoring only if after processing the text has zero positive as well as zero negative score, meaning either the text is neutral/is a fact or polarity scoring has failed to correctly classify the text. When such a case is encountered, the text/feedback is forwarded to the support vector classifier for classification.

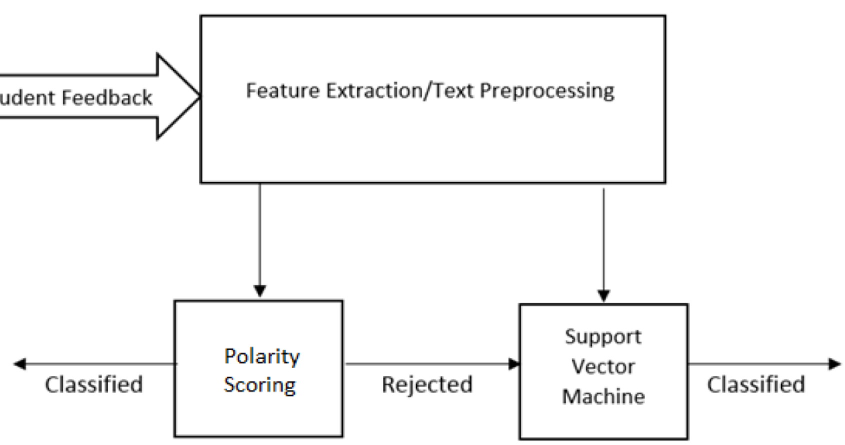


**Figure 3.** The architecture of the proposed two-stage classifier.

Some examples of such cases are given below where polarity lexicon fails but SVM succeeds in classifying: Example1. That was a great lecture.

Postprocessing text: great

Polarity Scoring Result: Positive polarity = 0.0 Negative Polarity = 0.0

SVM Result: Positive

Example2. The content is so unimaginative that our opinions are divided.

Postprocessing text: content

Polarity Scoring Result: Positive polarity = 0.0 Negative Polarity = 0.0

SVM Result: Negative

Example3. I think it is not okay as kids might not find it cool.

Postprocessing text: think

Polarity Scoring Result: Positive polarity = 0.0 Negative Polarity = 0.0

SVM Result: Negative

It is important to note that the support vector classifier does not come into play during each and every classification problem. It only handles difficult cases that polarity scoring is unable to handle properly. This approach uses the strengths of both polarity scoring and support vector classifiers. Since the present implementation of polarity scoring gives generally very good classification results, it is sufficient to classify most simple problems, resulting in small training and testing time periods unlike SVMs, which require large amount of data to train as well as a longer period of processing. Forwarding only select cases to the SVM classifier enables one to achieve a shorter duration of assessment and classification as it is not needed to send each and every case to the SVM classifier as most of these cases are easily handled by polarity scoring within relative shorter time periods. The overall classification algorithm is shown in Figure 4.

### 3.4. Experiments and results

The implementation and evaluation of several classifiers have been done by making use of the binary classifier strategy combined with the features defined earlier. The test data set has been taken from the Cornell Movie Review's dataset v2.0. This data set contains movie-review documents labelled with respect to their overall sentiment polarity or subjective rating as well as sentences labelled with respect to their subjectivity status, which is very useful for judging the acceptability of the content to its viewers. This made this data set structurally similar to educational content (e.g., lecture videos) reviews. While 65% of the dataset was used as training data, 35% of it was used as testing data. The experiments show that accuracy marks of close to 85% have been consistently achieved. The proposed combined strategy is named Intellimote Classifier in adherence to a tool with the same name, built for the purpose of training and testing along with creating a demo web-based application to exhibit the use of the strategy to good effect.

The test dataset is fed to the classifier as a single text file. Each review in the test dataset is subjected to different levels of preprocessing as described below to test the efficiency and effectiveness of the proposed preprocessing strategies.

- BINARY-LEX: Combination of binary strategy and basic tokenization referring to polarity lexicon.
- BINARY-LEX-RRC: Combination of binary strategy and basic tokenization referring to polarity lexicon. Repeated characters are removed as well.
- BINARY-LEX-NEG: Combination of binary strategy and basic tokenization referring to polarity lexicon. Negated words are considered and dealt with in appropriate fashion.

**Step 1:** Preprocess a text review to tokenize it into words, followed by removal of stop words and repeated characters, account for negation and tag each word/token with its corresponding parts of speech.

**Step 2:** Retain only tokens in the preprocessed review that are adverbs, adjectives, verbs, and are untagged, while discarding the rest.

**Step 3:** For each word in the processed list, check if it exists in the bag of words (SentiWordNet Database). If it does, add the word's positive and negative polarity score to the review's cumulative positive or negative polarity score. If the word has a negation tag, make sure to flip the polarity score of the word, i.e. if the word originally has a positive score add it to the negative score and vice versa.

**Step 4:** If the cumulative positive score is greater that the negative score, the review is deemed to be positive, otherwise it is deemed to be negative. If the scores are equal but not equal to zero, then it is deemed to be neutral.

**Step 5:** If the cumulative positive score and negative scores are both equal to zero then it is concluded that polarity scoring technique has failed in this case, and the review to a pretrained SVM classifier, which returns 'positive', 'negative', or 'neutral' as possible results.

**Figure 4.** The proposed classification algorithm.

- BINARY-LEX-STP: Combination of binary strategy and basic tokenization referring to polarity lexicon. All stop words are removed.
- BINARY-LEX-POS: Combination of binary strategy and basic tokenization referring to polarity lexicon. All tokens are POS tagged and tokens that are not verbs, adjectives, or adverbs are eliminated.
- BINARY-LEX-RRC-NEG-STP-POS: Combination of binary strategy and all other preprocessing levels/features.

Four classification strategies have been tested along with the preceding preprocessing strategies, namely naïve Bayes, maximum entropy, SVM and Intellimote. Results for all four classifiers applied along with the different preprocessing strategies were obtained and compared on the basis of the following parameters: accuracy, precision, recall, f-measure, and time. The results are presented in Figures 5–9, respectively. The performance of the four classifiers is ranked from best to worst as 1, 2, 3, and 4 and the performance of Intellimote in comparison to the other three standard classifiers has been analyzed. In the case of accuracy (refer to Figure 5), with BINARY-LEX, BNARY-LEX-RRC, and BNARY-LEX-NEG, the performance rank of SVM is 1, while that of Intellimote is 2. In the case of BINARY-LEX-STP, the performance rank of Intellimote is 4, while with BINARY-LEX-POS and BNARY-LEX-RRC-NEG-STP-POS Intellimote outperformed all the other three standard classifiers with rank 1. In the case of precision (refer to Figure 6), with four out of the six preprocessors, the performance rank of Intellimote is 2, while with the other two it is 1. In the case of recall (refer to Figure 7), with four out of the six preprocessors Intellimote's performance rank is 4, while with the other two the rank is 3 and 4, respectively. In the case of f-measure (refer to Figure 8), with five out of the six preprocessors, its performance rank is 2, while with the other one it is 1. In the case of time taken (refer to Figure 9), with all the six preprocessors, Intellimote took minimum time to complete the classification. It shows in all the cases the proposed classifier shows reasonably good results.

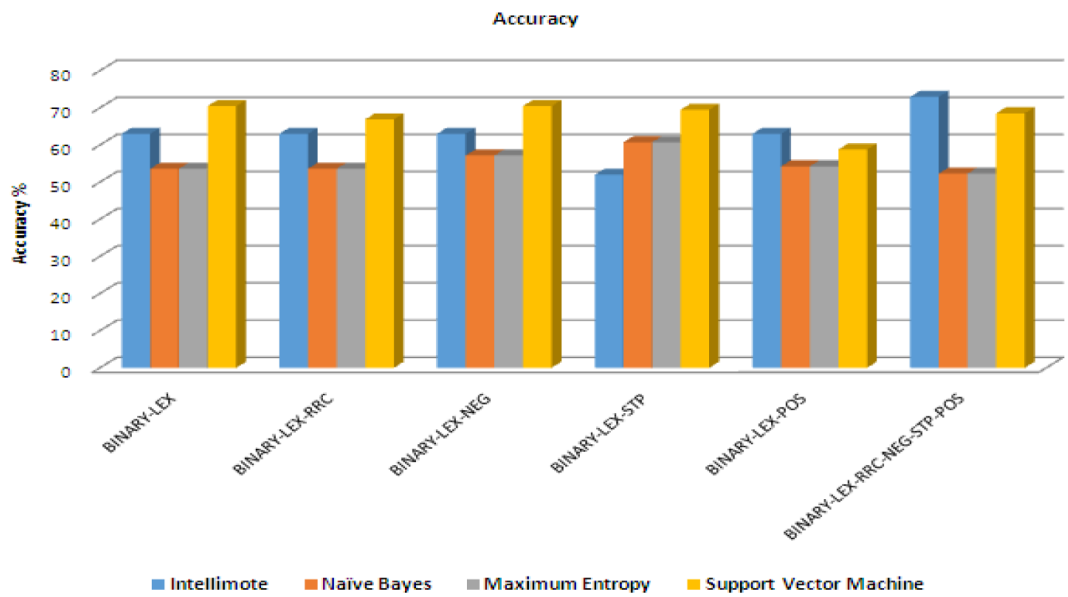


**Figure 5.** Comparison of accuracy with standard classifiers.

As mentioned earlier, the parameters measured for comparison are accuracy, precision, recall, f-measure, and time. The terms used to explain the parameters are defined below:

- TN (True Negatives): The number of true negatives, i.e. the no. of negative files that are analyzed properly.
- TP (True Positives): The number of true positives, i.e. the no. of positive files that are analyzed properly.
- FN (False Negatives): The number of false negatives, i.e. the no. of negative files that are wrongly analyzed.
- FP (False Positives): The number of false positives, i.e. the no. of positive files that are wrongly analyzed.

Accuracy: This is the ratio of the number of documents which are correctly analyzed to the total no. of documents and is calculated as

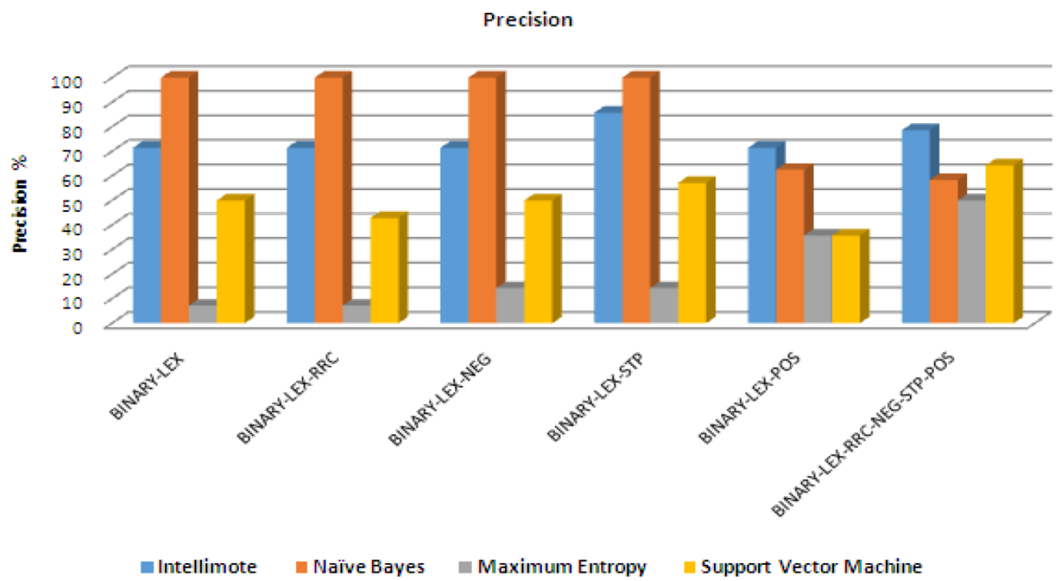


**Figure 6.** Comparison of precision with standard classifiers.

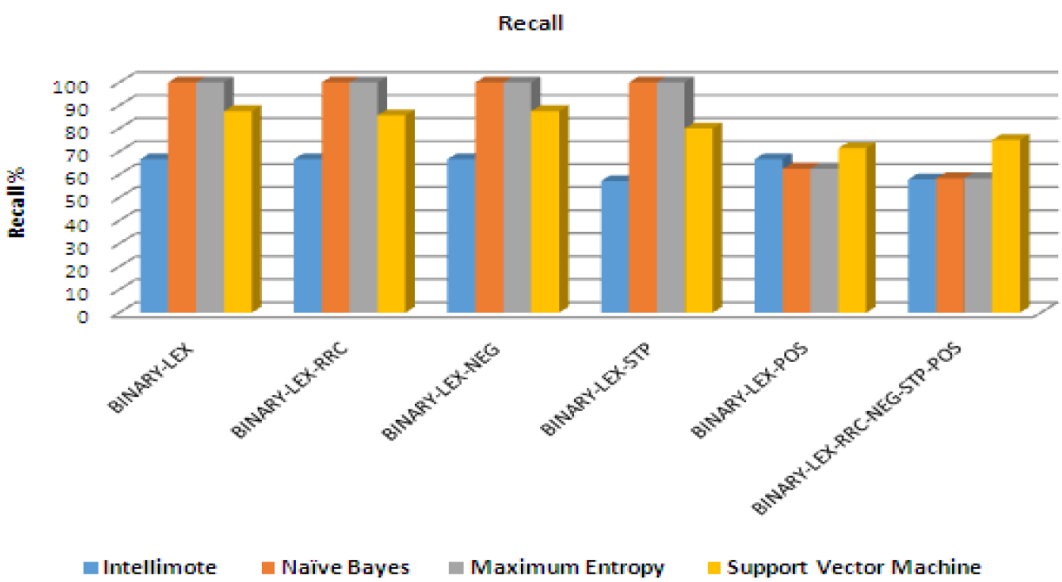


**Figure 7.** Comparison of recall with standard classifiers.

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

The total number of documents cannot be zero. At least some documents must be present. If total no. of documents is zero then accuracy cannot be measured.

Precision: Precision is the probability that a (randomly selected) retrieved document is relevant. This is the ratio of the total no. of positive files that are correctly analyzed to the total no. of true positive files and false positive files and is calculated as

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

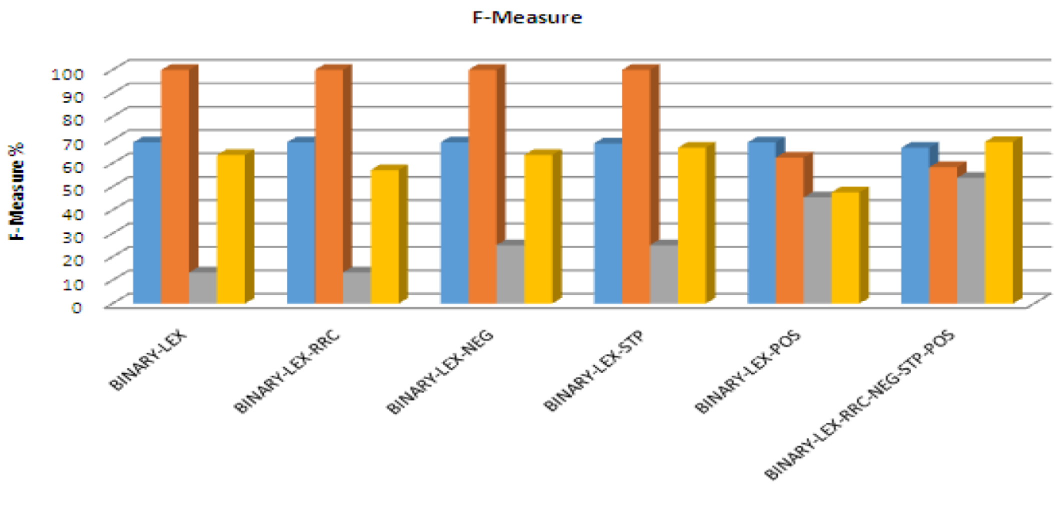


**Figure 8.** Comparison of f-measure with standard classifiers.

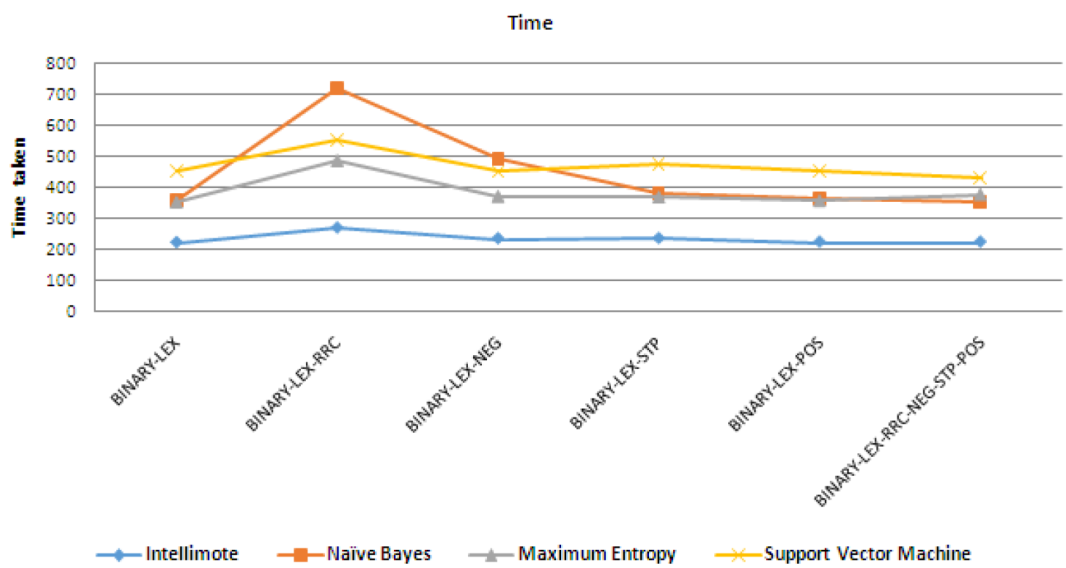


**Figure 9.** Comparison of time taken with standard classifiers.

The total number of positive files cannot be zero; else the calculated precision is zero.

Recall: Recall is the probability that a (randomly selected) relevant document is retrieved in a search. This is the ratio of the total no. of positive files that are correctly analyzed to the no. of positive files that are correctly analyzed summed with the no. of negative files that are analyzed incorrectly and is calculated as

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

Here the condition is that the TP + FN cannot be zero; else the calculated recall is zero.

F-Measure: This is the harmonic mean of precision and recall. The precision and recall cannot be equal to zero in order to do finite calculation and is calculated as

$$\text{F-Measure} = 2/((1/\text{Precision}) + (1/\text{Recall}))$$

Time: Total time taken to complete training and testing.

The entire backend of the application has been designed using Python and its accompanying distributions such as NLTK, NumPy, SciPy, and Scikit-Learn. Web py's web framework has been used to interface the python backend with the HTML, CSS, and JavaScript frontend.

## 4. Discussion and conclusion

The present authors have designed a sentiment analysis classifier using classifier combination rules, to combine polarity scoring and SVM in order to combine the benefits of both of these classifiers. As per experimental data, it is known that polarity scoring requires a much smaller amount of data to train and the training time is much less compared to other algorithms. If a substantial preexisting lexicon is used then training can be completely avoided. However, even though polarity scoring does a very good job, when it comes to basic classification every now and then it does encounter situations where it fails to classify a document correctly. To counteract these issues and to achieve a higher accuracy, a SVM is used to solve such document classification problems, which are otherwise difficult for polarity scoring.

SVMs generally have the highest accuracy of all classification algorithms but at the cost of a very large training dataset and much longer training time. By only forwarding selective tough problems to the SVM, the present work limits the amount of time spent on such problems, since most other problems are easily classified by polarity scoring. This way it designs an approach that gives the highest accuracy possible but with the lowest time spent in achieving that result.

The experiments show that accuracy marks of close to 85% have been consistently achieved. Another point to be noted is that the system is a self-adaptive system, i.e. it keeps learning from all the classification problems that are fed into it. With time it keeps updating its lexicon and keeps moving towards a higher accuracy level with minimal human intervention. This feature made it reasonably useful in the context of feedback analysis in different domains with a diverse and heterogeneous population of the learners of web-based e-learning environments.

In future, this work can be extended to try and achieve a much higher level of accuracy by implementing much more expansive text preprocessing algorithms wherein one can look into the possibility of dealing with negations in a POS-tagged manner as well as dealing with bigrams instead of unigrams, as certain word phrases or pairs often hold higher sentiment score compared to when considered independently.

## References

[1] Wang C, Xiao Z, Liu Y, Xu Y, Zhou A, Zhang K. SentiView: Sentiment analysis and visualization for internet popular topics. IEEE T Hum Mach Syst 2013; 43: 620-630.

[2] Pang B, Lee L. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2008; 2: 1-35.

[3] Turney PD. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: 40th Annual Meeting on Association for Computational Linguistics; 6 Jul 2002; Association for Computational Linguistics. pp. 417-424.

[4] Wilson T, Wiebe J, Hwa R. Just how mad are you? Finding strong and weak opinion clauses. In: AAAI 2004 21st Conference of the American Association for Artificial Intelligence; Jul 25 2004; San Jose, CA, USA. AAAI. vol. 4. pp. 761-769.

[5] Baroni M, Vegnaduzzo S. Identifying subjective adjectives through web-based mutual information. In: KONVENS; Sep 2004. vol. 4. pp. 17-24.

[6] Turney P, Littman ML. Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Technical report NRC technical report ERB-1094. Institute for Information Technology, National Research Council, Canada. 2002.

[7] Gamon M, Aue A. Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms. In: ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing; Jun 29 2005. pp. 57-64.

[8] Pang B, Lee L, Vaithyanathan S. Thumbs up?: sentiment classification using machine learning techniques. In: ACL 2002 Conference on Empirical Methods in Natural Language Processing; Jul 6 2002. vol. 10. pp. 79-86.

[9] Prabowo R, Thelwall M. Sentiment analysis: A combined approach. J. Informetrics. 2009; 3: 143-157.

[10] Shein KP. Ontology based combined approach for sentiment classification. In: 3rd International Conference on Communications and information technology; 29 Dec 2009; World Scientific and Engineering Academy and Society (WSEAS). pp. 112-115.

[11] Colbaugh R, Glass K. Estimating sentiment orientation in social media for intelligence monitoring and analysis. In: IEEE 2010 International Conference on Intelligence and Security Informatics (ISI); 23 May 2010. pp. 135-137.

[12] Esuli A, Sebastiani F. Sentiwordnet: A publicly available lexical resource for opinion mining. In: LREC 2006; 22 May 2006. vol. 6. pp. 417-422.

[13] Wang H, Xu H, Liu L, Song W, Du C. An unsupervised microblog emotion dictionary construction method and its application on sentiment analysis. Journal of Information & Computational Science 2015; 12: 2729-2739.

[14] Stone PJ, Dunphy DC, Smith MS. The General Inquirer: A Computer Approach to Content Analysis. Cambridge, MA, USA: MIT Press, 1966.

[15] Whissell C. Using the revised dictionary of affect in language to quantify the emotional undertones of samples of natural language. Psychol Rep 2009; 105: 509-21.

[16] Miller G, Fellbaum C. Wordnet: An Electronic Lexical Database. Cambridge, MA, USA: MIT Press, 1998.

[17] Poria S, Gelbukh A, Cambria E, Hussain A, Huang GB. EmoSenticSpace: A novel framework for affective common-sense reasoning. Knowledge-Based Systems 2014; 69: 108-123.

[18] Li W, Xu H. Text-based emotion classification using emotion cause extraction. Expert Syst Appl 2014; 41: 1742-1749.

[19] Bravo-Marquez F, Mendoza M, Poblete B. Meta-level sentiment models for big social data analysis. Knowledge-Based Systems 2014; 69: 86-99.

[20] Jian C, Gao J, Ao Y. A new sampling method for classifying imbalanced data based on support vector machine ensemble. Neurocomputing 2016; 193: 115-122.

[21] Peker M. A decision support system to improve medical diagnosis using a combination of k-medoids clustering based attribute weighting and SVM. J Med Syst 2016; 40: 1-16.