UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

Sentiment Analysis of Student's Opinion on Programming Assessment: Evaluation of Naïve Bayes over Support Vector Machines

Mahmood Umar
Department of Computer Science,
Faculty of Science, Sokoto State University,
P.M.B 2134, Birnin Kebbi Road (Near Airport)
Sokoto State Nigeria
mahmoodumar24@gmail.com

Nor Bahiah Ahmad
School of Computing,
Faculty of Engineering,
Universiti Teknologi Malaysia,
81310 UTM Johor Bahru,
Johor, Malaysia
bahiah@utm.my

Anazida Zainal
School of Computing,
Faculty of Engineering,
Universiti Teknologi Malaysia,
81310 UTM Johor Bahru,
Johor, Malaysia
anazida@utm.my

Submitted: 16/09/2020. Revised edition: 09/11/2020. Accepted: 10/112020. Published online: 19/11/2020
DOI: https://doi.org/10.11113/ijic.v10n2.278

***Abstract*—This study investigates the performance of machine learning algorithms for sentiment analysis of students' opinions on programming assessment. Previous researches show that Support Vector Machines (SVM) performs the best among all techniques, followed by Naïve Bayes (NB) in sentiment analysis. This study proposes a framework for classifying sentiments, as positive or negative using NB algorithm and Lexicon-based approach on small data set. The performance of NB algorithm was evaluated using SVM. NB and SVM conquer the Lexicon-based approach opinion lexicon technique in terms of accuracy in the specific area for which it is trained. The Lexicon-based technique, on the other hand, avoids difficult steps needed to train the classifier. Data was analyzed from 75 first year undergraduate students in School of Computing, Universiti Teknologi Malaysia taking programming subject. The student's sentiments were gathered based on their opinions for the zero-score policy for unsuccessful compilation of program during skill-based test. The result of the study reveals that the students tend to have negative sentiments on programming assessment as it gives them scary emotions. The experimental result of applying NB algorithm yields a prediction accuracy of 85% which outperform both the SVM with 70% and Lexicon-based approach with 60% accuracy. The result shows that NB works better than SVM and Lexicon-based approach on small dataset.**

***Keywords*—Sentiment Analysis, Programming Assessment, Naïve Bayes, Support Vector Machines and Lexicon-Based Approach**

I. INTRODUCTION

Sentiment Analysis, known as Opinion Mining is the computational study of people's opinions, attitudes, and emotions towards an entity [1]. The entity can represent individual events or topics in the following area, such as education, e-commerce, health, politics and many more. This research centered on education specifically, analysis of student sentiments on programming assessment using data mining techniques in sentiment analysis. Performance evaluation of Naïve Bayes (NB) algorithm over Support Vector Machines (SVM) and Lexicon-based approach is to be conducted in order to find out the efficiency of machines learning algorithms on small data set. Hence, the need to investigate the performance of these machine learning algorithms on small data is the concern in this study.

School of Computing, UTM offers programming courses namely Programming Technique I and Programming Technique II. The subjects are core courses compulsory for the first-year undergraduate students. The mode of assessment is an issue bothering the students as zero score is awarded to student with unsuccessful compiled program. Consequently, 146 respondents filled the survey but only 75 comments were suitable to be collected from the students as corpus data for the sentiment analysis. This is the reason why the data is considered as small, but sufficient to test machine learning algorithms which ideally, work better on large data sets. The study will design a hybrid framework to classify student's sentiment as positive or negative regarding the programming assessment conducted in School of Computing, UTM.

II. BACKGROUND OF THE STUDY

Machine learning algorithms, specifically Naïve Bayes algorithm was used by [2] in a study to conduct sentiment analysis of first-year engineering courses based on student feedback. The author used 1000 data in his experiment. In another study by [3], the authors conclude that Naïve Bayes, Maximum Entropy and SVM are three classifiers that have superior performance for sentiment analysis. These algorithms had effective performances, and SVM gives the best result. In [4], the authors designed a sentiment analysis model for Anadolu University using Naïve Bayes classifier specifically opinion finder software to analyse student's opinion collected from twitter. The processes of sentiment analysis differ based on the type of classes to predict (positive or negative, subjective or objective) and different levels of classification (sentence, phrase, or document level and language). The authors used opinion finder software as a tool in the sentiment analysis. The issue with this software is that sentiment lexicon was mainly used for subjectivity finding in a sentence. It is a tool used by the Lexicon-based approach in sentiment analysis. Therefore, it is not suitable for the sentiment classification for machine learning.

III. RELATED WORKS

Sentiment analysis has been conducted on several entities. Different entities determine the type of techniques, approaches or tools to be used. It also determined the type of architecture, model or framework to be designed by the researcher. This section review some of the techniques, architectures and frameworks developed by the researchers in various area of applications in sentiment analysis.

Fig. 1 is the framework used by [5] in the sentiment analysis of twitter data for stock market price prediction using Naïve Bayes algorithm. The framework consists of five phases which are data collection, data pre-processing phase, expert labelling, hybridization of Naïve Bayes classifiers, and performance evaluation and result. The authors used Naïve Bayes in the framework. Hence, there is a need to design a hybrid framework involving three machine learning algorithms such as NB, SVM and Lexicon-based to get a better performance algorithms.

Fig. 2 is the optimized sentiment analysis framework(OSAP) for SVM used by [6]. The authors analyzed customers' opinions on products from micro blogging websites. It is made up of four phases namely: data set collection, data pre-processing, sentiment classification, and result evaluation. The framework was designed based on SVM and does not have evaluation phase for the hybrid approach.

The author in [7] classify student's sentiment as either positive or negative for the improvement of teaching and learning in an Open Arab University Business Program Courses Case Study. The framework introduced by the author is made up of five steps which are, data collection that contain sentences which are either positive sentences or negative sentences, dividing the data into training and testing set, learn classifier(subjectivity) and performance evaluation. The authors categorized the words as good, awesome, bad and awful and used certain weight for each category.

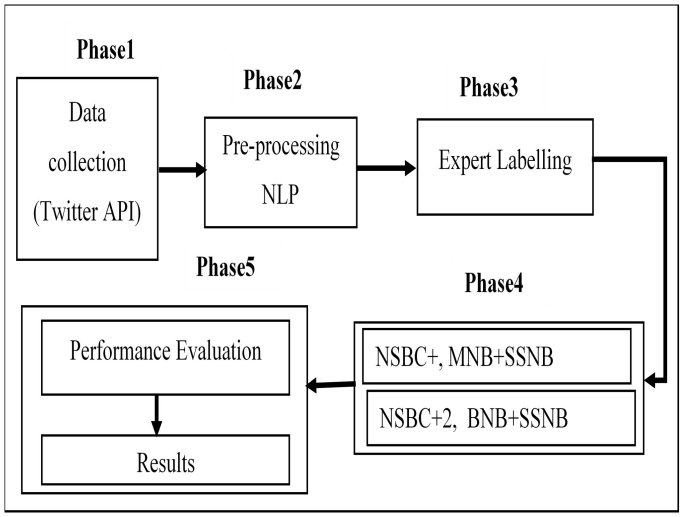

Fig. 1. Sentiment Analysis Conceptual Framework [5]

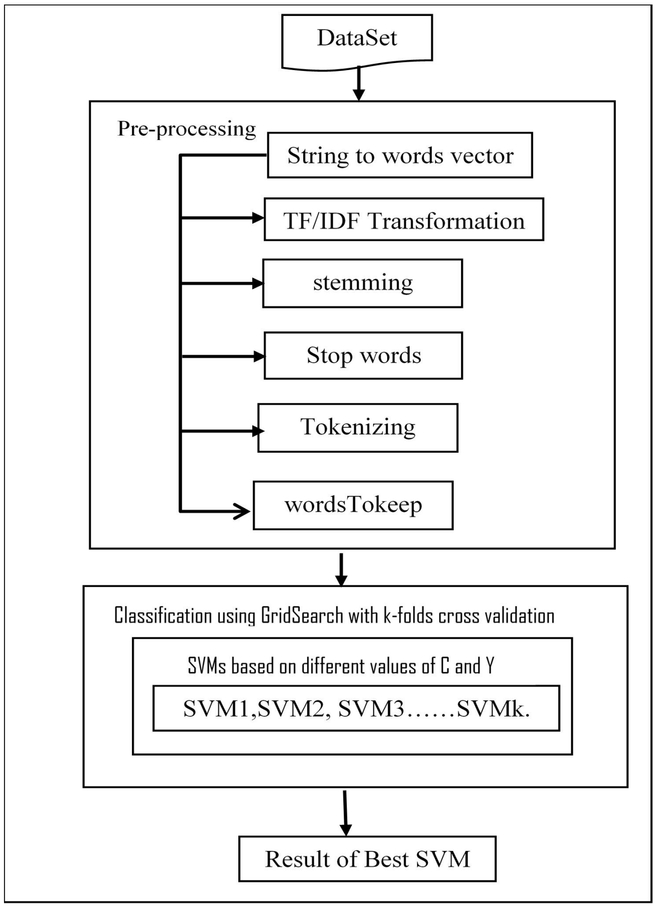

Fig 2. Optimized Sentiment Analysis Framework (OSAP) for SVM [5]

Fig 3 shows the NRC Lexicon sentiment analysis architecture used by [8] and [9] for the improvement of teaching and learning. The authors collected students' feedback on teaching methodology and course satisfaction. There are five main phases in the architecture: data collection, data pre-processing, sentiment and emotion identification, satisfaction and dissatisfaction computation and result visualization. The data collected from students via formal source using survey and informal sources from blogs and forums were pre-processed. The pre-processed data were passed to the next phase for sentiment and emotion identification. The sentiments are classified as either positive or negative. The emotions were classified based on eight categories; anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. The proposed system computes satisfaction or dissatisfaction based on these labelled emotions. The issue with this architecture is the scope. The system processed the collected data in multilingual based on two entities. Therefore, this is a motivational factor for this study to focus on one entity, programming assessment only and one language, English.

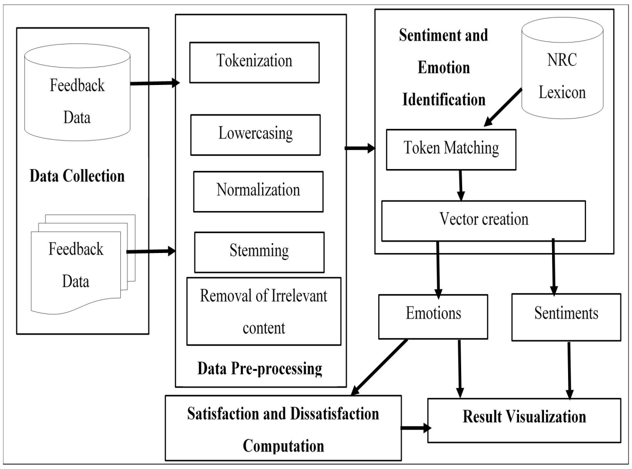

Fig. 3. NRC Lexicon Sentiment Analysis System Architecture [8]and [9]

According to [10] there is an increase in the number of researches on the application of machine learning algorithms in sentiment analysis. The author identified Naïve Bayes, Random Forest, Decision Tree, Neural Networks and K-Nearest Neighbor as the commonly used techniques in this area. The study also shows that the best performing machine learning algorithms on large data are SVM and NB algorithm. SVM outperforms NB because of its ability to analyze non-linear data. This is due to the presence of a kernel that forms a hyperplane from the data. The data to be used in this study is textual which is linear. Therefore, the performance of SVM over NB is not guaranteed. In addition to that, SVM outperforms NB because the data is large, the case may be the different on small data. This study also intends to use the Lexicon-based approach to test the performance of NB. SVM is the machine learning algorithm chosen to validate NB because it is found to be the best in sentiment classification.

IV. METHODOLOGY

This section describes the procedures followed in designing the research framework for the sentiment analysis. The framework comprises of four phases, which are: Online Survey and data preparation, Data pre-processing, Sentiment classification and Performance Evaluation. Fig. 4 outlines the framework proposed in this study.

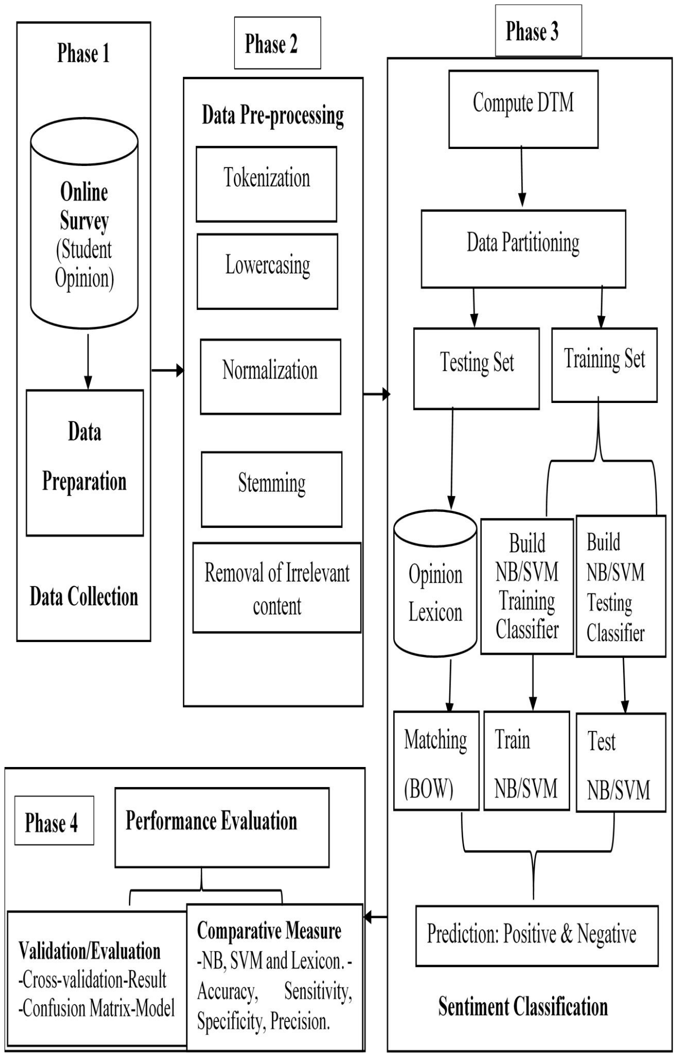

Fig. 4. Proposed Sentiment Analysis Framework

A. Proposed Sentiment Analysis Framework

The framework in Fig. 4 is a combination of different frameworks used by the researchers in sentiment analysis as described in Section II. The framework consists of four phases. The first phase is data collection which includes online survey and data preparation. Phase 2 is the data preprocessing phase, phase 3 is the sentiment classification and phase 4 is the performance evaluation which consists of result validation (cross-validation technique) and comparative measure.

i. Phase 1: Data Collection (Online Survey and Data Preparation)

In this phase, the data is collected among first year undergraduate students in School of Computing, UTM. Learning programming is a core course for the first-year students taking Bachelor of Science (Computer Science) offered at the School of Computing, UTM. There are five study programmes, which are Bioinformatics, Software Engineering, Computer Network and Security, Graphics and Multi-Media Software and Data Engineering. Students involved in this study are registered in Programming Technique II (SCSJ1023) subject which they already fulfill Programming Technique I (SCSJ1013) subject as prerequisite. To ensure that student not only master the theoretical part of the programming, skill-based tests were part of the assessment that test the programming skill amongst students.

The data was collected using an online survey via nine WhatsApp groups; representing nine sections or groupings for Programming Technique II classes. In the online survey, there are General Comment sections that need to be filled by the students and the responses were considered as sentiment data. Only 75 respondents out of 146 submitted their opinions on skilled-based test 1 and skilled-based test 2. Even though, this data is considered as small, but it is sufficient to carry out the experiment for this research. The data was extracted as textual in excel sheet, and each comment was labelled manually as positive and negatives sentiments in the excel. This is called data preparation.

After the data is collected, the next step is data preparation as shown in the framework in Fig. 4. It involves extraction and preparation for import into the RStudio compiler for pre-processing of the data. The extraction requires that the raw data from the online survey being exported as Excel datasheet. Machine learning requires that the data to be labelled as positive or negative based on the text orientation. The data was labelled manually with the number of positive and negative sentiments are 44 and 31 respectively. However, the Lexicon-based approach works directly on the text data.

ii. Phase 2: Data Pre-processing

The second phase is the pre-processing which involves tokenization, stemming, white space, stop words, and irrelevant content removal. Fig. 5 shows the stages in the data preprocessing. All procedures were implemented using libraries and methods in R compiler.

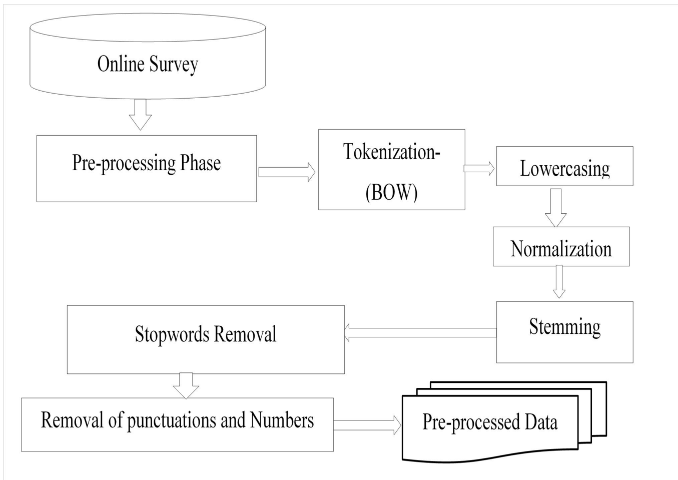

Fig. 5. Data Preprocessing Stages

Fig. 6 and 7 shows the sample of the data before and after preprocessing respectively. All the stages were implemented using the lexicon-based approach as shown in Fig. 5. The tokenization of the text is implemented by separating each word in the sentences with a space. This implies that unigram type of tokenization is used. The spaces were removed using the function “strip_ whitespace ()” in the pre-processing. It can be seen that all the stop words in the English language were removed. For example, articles like “the”, “an”, “a”, conjunction like “and”,” or” and so on. Punctuation marks, digits and symbols are also removed (e.g. increment operator in C++). It can also be observed that all the text shown in Fig. 7 has been converted to the lowercase by the R compiler.

```
[1] pleased worst decision ever taken coordinator scsj prepare exams honest
find questions straightforward problem one compile program automatically get
 zero mark absentmindedly taken decision ever since point studying c hard ye
t know code due stringent rules examining one tend loose reminiscing hope to
wards course reflects utter laziness lecturers

[2] good test however giving zero marks program compile little harsh

[3] better similar examples given students wont panick

[4] well machine maintain better everytime test Skill-based test finals mach
ine acts middle process either crash irrelevant error im

[5] unfair program run get zero

[6] second question final hard dont even understand question question style
different past year question
```

Fig. 6. Sample of Data Before Preprocessing

```
[7]im lacking exercise
[24] tough
[1] rule change getting zero programs compile re evaluated fair people
get marks parts get right change policy led many students just leaving
things afraid try new ways solve problem afraid compile will get zero
settled less marks rather none
still marks eventhough succesfully compile
[5] unfair program run get zero
[13] second question final hard dont even understand question question
style different past year question
[35] ok overall
[3] good us
[4] hardwork needed
[19] Skill-based test great test test ability skills however hope
practices different difficulties can done well prepare
[11] final question difficult
[46]Answer need compile otherwise get zero marks questions difficult
help uss hate programming
```

Fig. 7. Sample of Data After Preprocessing

iii. Phase 3: Sentiment Classification

In the third phase, the pre-processed data undergoes sentiment classification using Naïve Bayes algorithm, Lexicon-based approach and SVM as shown in Fig. 4. The data is classified as positive or negative. The stages under this phase are representation of a bag of words known as document term matrix, data partitioning, and building and training Naïve Bayes classifier algorithm.

- Document Term Matrix: This implies that the rows of the document term matrix equal to the corpus (documents in the collection) and columns refer to the terms, and its elements are the term frequencies. The implementation of the document term matrix is done by importing the text mining package called "tm" in R compiler.
- Data Partitioning: For the machine learning algorithm like Naïve Bayes, the data set is divided into two training data set and testing data set based on percentage and ratios for sentiment classification. Based on the 75 feedbacks collected from the students, the data partition is done in two ratios 55:20 and 50:25 represented by 66% vs 44% and 73% vs 27% for both SVM and NB in training and testing sets respectively. In the case of the Lexicon (which is not a machine learning approach), the experiment is implemented using 25 and 20 ratios as testing sets because the algorithm does not require training set.
- Building and Training Classifier: In order to implement Naïve Bayes algorithm in R compiler, the libraries in RStudio package called "e1071" were already installed and loaded for the implementation. This procedure is only applicable to Naïve Bayes and SVM because they require training and testing data set. The Lexicon-based approach works directly on the pre-processed data. The detailed implementation for both the machine learning algorithms and the lexicon-based approach will be explained in Section V.

iv. Phase 4: Performance Evaluation

The fourth phase of the framework is performance evaluation as shown in Fig. 3 in order to measure the accuracy of Naïve Bayes algorithm, the data was divided into two; training and testing sets, this is called cross-validation or data partitioning.

Evaluation metrics or F-measures are the performance measures used to evaluate the accuracy of an algorithm. These include sensitivity, specificity, positive predicted value, accuracy rate, error rate etc. They are computed based on the output of the confusion matrix viz; True Positive (TP), False Positive(FP), True Negative (TN), and False Negative(FN) [14]. Confusion matrix is shown in Table 1 whereas the procedure for the performance evaluation is summarized in in Table 2.

TABLE 1. Confusion Matrix

		ACTUAL CLASS	
		Negative	**Positive**
PREDICTED CLASS	**Positive**	True Negative (TN)	False Negative (FN)
	Negative	False Positive (FP)	True Positive (TP)

TABLE 2. Performance Evaluation Measures and Procedures

Parameter	Description and Procedure
Accuracy	Overall percentage accuracy of the sentiment's prediction is calculated as: Accuracy=TP+TN/(TP+TN+FP+FN)
Confidence Interval-CI	The range (minimum and maximum prediction accuracy) within which the accuracy of the model is expected at a 95% confidence interval. Minimum=0.5061 and Maximum= 0.8793.
No information	It is the knowns as Best guess based on the majority class. Negative class is the majority and accuracy achievable by predicting this class is always 60%,
P-Value [Acc > NIR]	The probability value is good when accuracy is greater than the No information rate.0.72>0.6
Kappa	It indicates how good the algorithm predictions match with actual labels. Kappa= 0.3636
McNemara's Test P-Value	Determines whether row and column marginal frequencies are equal
Sensitivity or Recall	Sensitivity is also known as recall or true positive rate. SN= TN/(TN+FP) or TN/N
Specificity	Specificity means true negative rate.TN= TP/(TN+FN) or TP/P
Positive Pred. Value	This is also called precision. It is computed as; Positive Predictive Value= TN/(TN+FN)

Parameter	Description and Procedure
Negative Pred. Value	Negative Prediction Value= TP/(TP+FP)
Prevalence	Indicates how frequent the positive class occur in the data sample. P=(TN+FP)/(TP+FP+FN+TN)
Detection rate	It refers to the sum of correct positive class predictions made as a proportion of all predictions. DR=TN/(TP+FP+FN+TN)
Detection Prevalence	Refers to the sum of all negative class as a proportion of all predictions. DP=(TN+FN)/(TP+FP+FN+TN)
Balance Accuracy	Refers to average of true positive and true negative rates. Balance Accuracy=(Specificity + Sensitivity)/2
Error Rate	It is the sum of the false positive and negative prediction made as a proportion of all prediction. Error Rate=(FP+FN)/ (TP+FP+FN+TN)

V. EXPERIMENT

This section presents the implementation of the proposed framework by conducting three experiments with Naïve Bayes, Lexicon and SVM, results analysis, discussions and summary. All the experiments are conducted using R compiler (RStudio and RTool).

After the data pre-processing phase, the first experiment was done using NB algorithm following the flowchart used by[15]. The necessary packages need to be loaded into the R programming compiler. The compilers used are RStudio version 3.6.1 and RTool version 1.2.5.1. The packages were used to implement the pre-processing stages and the NB algorithm. The next stage of the experiment is computing the document term matrix (DTM) of the corpus data. DTM is the matrix representation of the bag of the word (BOW). A built-in method in the 'tm' library of RStudio compiler is used to compute the DTM.The sixth stage is data partitioning. This stage is only applicable to the machine learning algorithms- Naïve Bayes and SVM. It was described in section III.

The result of the partition is in 2 ratios; The first one is 50:25 represented by 66% and 44% of the data while the second ratio is 55:20 represented by 73% and 27% of the data. It can be observed that more samples were allocated to training because the data is small. And this decision yields a better result than allocating fewer samples to the training set. The seventh stage is training the machine-learning algorithm using the 2 ratios described in the fourth stage. The library e1071 contains a function that builds two classifiers that work on the data partitions: training and testing set. The function that is used to train the Naïve Bayes is "dim(dtm.train.nb)". And the one for testing the Naïve Bayes is implemented using "dim(dtm.train.nb)". The Naïve Bayes algorithm converts the term frequencies from digits (0 or 1) to Boolean values (YES or NO) which imply presence or absence.

The following stage is sentiment prediction by the machine learning algorithms. The algorithms (NB and SVM) use a function called a confusion matrix to present the predicted number of positive and negative sentiments. The two ratios (50:25 and 55:20) are considered in the experiment in both NB and SVM. The result of the experiments is summarized in the confusion matrix in Table 3 and Table 4.

The second experiment is the Lexicon-based approach specifically, BOW model. The aim is to test the NB with an approach different from the machine learning algorithm for performance comparison on the small data.

The implementation of the first stage up to the sixth stage of the Lexicon-based approach is the same as described in the NB algorithm. The only distinction is as follows: The data used in the experiment is text. The packages loaded into RStudio compiler at the fourth stage are text mining "tm" and "stringr". The results obtained at stage five is shown in Section III At stage six, the result after computing document term matrix is shown in Table 3. The seventh stage is loading the BOW. This is a dictionary of positive and negative sentiments used by [16]. It was downloaded and saved into the appropriate directory for the purpose of this research. It contains sentiments as positive-2006 and negative-4783 terms. In a study by [16] BOW is used in sentiment analysis of customer review data. At the eighth stage, the Lexicon method in RStudio compiler is loaded so that it can scan through the BOW for the positive and negative sentiment. At the ninth stage, the lexicon compares between BOW or corpus data with the opinion lexicon(dictionary) and returns the matches found as the predicted results. The result is the predicted as positive and negative sentiments.

Experiment three involve the use of SVM. The main reason for conducting the experiment is to validate the proposed framework with another machine learning algorithm that is known to be good in sentiment classification. As mentioned earlier, SVM is reported to be the best in terms of sentiment classification on large data. It outperforms the NB, lexicon and other machine learning algorithms. The experiment for SVM is the same as the stages used in the NB experiment. The only difference is at the fifth stage where the training and testing are done by the SVM algorithm. In addition to that SVM convert text data to a vector and find a hyperplane between the two classes; negative and positive using a specialized function called kernel[17]. It is implemented by caret package in RStudio compiler. The kernel is used by the SVM in pattern analysis. It is mainly used when the data is non-linear. Since the study used text data in excel datasheet which is a linear type of data structure, there is no need for the kernel.

VI. RESULT

This section presents the result of the three experiments using NB, Lexicon-based approach and SVM conducted based on 50:25 and 50:20 training/testing sets respectively, for all the three algorithms from the confusion matrices generated by R studio compiler and was summarized in Table 3. From the table, it can be seen that with the ratios 50:25, NB has prediction accuracy of 72% and outperformed SVM with 68% and Lexicon with 60%. While on the ratio of 50:20, NB with prediction accuracy 85% outperformed SVM with 70% and Lexicon with 50%. The corresponding predictive positive sentiment based on 50:25 ratio, NB has 70% which outperformed SVM with 0% and Lexicon with 50%. Based on 50:25% SVM has 100% negative sentiments prediction which outperformed NB with 80% and lexicon with 66%. On the positive prediction SVM is the best. Based on 50:20 ratio, the SVM had 92% negative prediction which outperformed NB with 75% and lexicon with 60. On the positive prediction, NB has 75% which outperform SVM with 0% and lexicon with 60%.

VII. DISCUSSIONS

Table 3 and Table 4, show that, NB has the overall prediction accuracy of 72% and 85% on the ratio 50:25 and 50:20 respectively. While SVM has a prediction accuracy of 65 and 70% on the ratio 50: 25 and 50:20 respectively. Lexicon is the least in terms of prediction accuracy with 60% and 50% on 50:25 and 50:20 rations respectively. As mentioned in section V, the result of classifying sentiment using all the three (3) algorithms shows that, NB has more negative prediction of 85% over positive prediction. So, therefore the conclusion is that the overall student sentiment on Skill-based test is negative with an accuracy of 85% using NB. Hence skill-base test gives negative emotions to the first-year students of the School of Computing. The Skill-based test may trigger the student to dislike C++ programming.

TABLE 3. Results Comparison-NB, SVM and Lexicon-Based Approach

	Training/Testing (50:25)			**Training/Testing (50:20)**		
Algorithms	Accuracy %	Pos Pred. %	Neg Pred. %	Accuracy %	Pos Pred. %	Neg Pred. %
NB	72	70	80	85	75	92
SVM	68	0	100	70	0	100
LEXICON	**Testing:25**			**Testing:20**		
	60	50	66	50	60	60

TABLE 4. Statistics Measures for The Performance Evaluation of NB and SVM

Parameter	**NB-50:25**	**NB-55:20**	**SVM-50:25**	**SVM-55:20**
Accuracy	0.72	0.85	0.68	0.7
95% CI	0.5061- 0.8793	0.6211-0.9671	0.465-0.8505	0.4572-0.8811
No Information Rate	0.6	0.65	0.68	0.7
P-Value [Acc > NIR]	0.1536	0.04438	0.59428	0.60801
Kappa	0.3636	0.6809	0	0
Mcnemar's Test P-Value	0.1306	1.00000	0.01333	0.04123
Sensitivity	0.9333	0.8462	1.0	1.0
Specificity	0.4000	0.8571	0.00	0.0
Pos Pred Value	0.7000	0.7500	0.68	0.7
Neg Pred Value	0.8000	0.9167	0	0
Prevalence	0.6000	0.6500	0.68	0.7
Detection Rate	0.5600	0.5500	0.68	0.7
Detection Prevalence	0.8000	0.6000	1.0	1.0
Balanced Accuracy	0.6667	0.8516	0.50	0.5
'Positive' Class	Neg	Neg	Neg	Neg

Table 4 shows the details result for performance evaluation metrics of machine learning algorithm and the metrics were used by [18] to evaluate the result of a study. The table shows that the Naïve Bayes algorithm did well in the prediction of the sentiments with an accuracy of 72% and the class interval of 50% to 88% which is a good range for lower and upper accuracy respectively. The p-value is good when the accuracy is greater than no information rate. From the table we can compute p-value as accuracy >no information rate (i.e., 0.72>0.6, the difference is 0.1536). No information rate is 60% which is the best guess beyond the overall distribution of the classes. That is to say, 60% of the respondent have negative opinions on the skill-based test. Recall that, the number of positive and negative sentiments from the sample data after the labelling is 44 and 31 (as discussed in section III) respectively which support our prediction, and hence yielding the No information rate of 60%. The kappa value is 0.3636 or 36%. It is the comparison between the observed and expected accuracy. The McNemar's Test P-value is used to determine whether row and column marginal frequencies are equal and the value is 0.1306.

According to Table 4, sensitivity has a percentage value of 93% while specificity has 40% respectively. The table also shows that the predicted positive values in percentage is 70% while the predicted negative values in percentage is 80% with a prevalence value of 60%. The detection rate is 56% with a prevalence of 80%.

The overall prediction results show that NB performs better than the Lexicon-based approach on small data set with 85% accuracy. SVM has 70% prediction accuracy; better than lexicon-based approach with 60% prediction accuracy due to small data. To solve this issue of insufficient data for the machine learning algorithms, large data can be obtained from online data source such as Machine learning repositories (UCI), Kaggle, Amazon, etc.

In order to improve this research, three (3) suggestions were offered viz: the experiment in section V can be repeated using a big data set in a different language- Malay language. Students from more than one level, course of study and course syllabus should be considered in order to implement feature selection for the machine learning algorithm which limits frequency of the words, for a better result. More features on many skill-based tests may also emerge that affect student performance assessment. The experiment (in section V) should also be done using other ML algorithms for sentiment analysis-Maximum Entropy, Random Forest and Decision Tree for effective performance evaluation of NB algorithm. Thirdly, the data can be collected from various social media platform such as WhatsApp, Telegram, Facebook and twitter in order to test NB algorithm performance over other algorithms on different data set. This will also explore additional features and metrics in sentiment classification on the skill-based test sentiment analysis.

Acknowledgment

The authors are grateful to the lecturers and students at School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia for supplying the data for this study. Their responses from the survey have been pre-processed and analyzed in this study.

REFERENCES

[1] W. Medhat, A. Hassan, and H. Korashy. (2014). Sentiment Analysis Algorithms and Applications: A Survey. *Shams Eng. J.*, 5(4), 1093-1113.

[2] E. Andersson, C. Dryden, and C. Variawa. (2018). Methods of Applying Machine Learning to Student Feedback through Clustering and Sentiment Analysis. *Proc. Canadian Engineering Education Association(CEEA-ACEG18) Conf.*, 2-7.

[3] N. Altrabsheh, M. M. Gaber, and M. Cocea. (2013). SA-E: Sentiment Analysis for Education. *Frontiers in Artificial Intelligence and Applications*, 255(June), 353-362.

[4] Z. K. Ozturk, Z. İ. E. Cicek, and Z. Ergul. (2017). Sentiment Analysis : An Application to Anadolu University.

[5] G. A. A. J. Alkubaisi, S. S. Kamaruddin, and H. Husni. (2018). Conceptual Framework for Stock Market Classification Model Using Sentiment Analysis on Twitter Based on Hybrid Naïve Bayes Classifiers. *Int. J. Eng. Technol.*, 7(2), 57-61.

[6] M. Ahmad, S. Aftab, M. S. Bashir, N. Hameed, I. Ali, and Z. Nawaz. (2018). SVM Optimization for Sentiment Analysis. 9(4).

[7] M. Atif. (2018). An Enhanced Framework for Sentiment Analysis of Students' Surveys: Arab Open University Business Program Courses Case Study, 9(1), 9-11.

[8] R. Archana and B. Kishore. (2017). Role of Sentiment Analysis in Education Sector in the Era of Big Data: A Survey. *Int. J. Latest Trends Eng. Technol.*, 022-024.

[9] S. Rani and P. Kumar. (2017). A Sentiment Analysis System to Improve Teaching and Learning. *Computer*, 50(5), 36-43.

[10] A. Lajis, S. A. Baharudin, D. A. Kadir, N. M. Ralim, H. M. Nasir, and N. A. Aziz, (2018). A Review of Techniques in Automatic Programming Assessment for Practical Skill Test. *J. Telecommun. Electron. Comput. Eng.,* 10(2-5): 109-113.

[11] M. Abbas, K. Ali Memon, and A. Aleem Jamali. (2019). Multinomial Naive Bayes Classification Model for Sentiment Analysis. *IJCSNS Int. J. Comput. Sci. Netw. Secur.*19(3), 62.

[12] M. Ahmad, S. Aftab, S. Muhammad, and S. Ahmad. (2017). Machine Learning Techniques for Sentiment Analysis: A Review. *Int. J. Multidiscip. Sci. Eng*, 8(3): 27-32.

[13] B. M. Jadav and M. E. Scholar. (2016). Sentiment Analysis using Support Vector Machine based on Feature Selection and Semantic Analysis, 146(13), 26-30.

[14] N. Zainuddin and A. Selamat. (2014). Sentiment Analysis using Support Vector Machine," *I4CT 2014 - 1st Int. Conf. Comput. Commun. Control Technol. Proc.*, May 2016, 333-337.

[15] R. Shukla and N. Mishra. (2016). Framework for Sentiment Analysis of Twitter Post, 3426-3431.

[16] M. Hu and B. Liu. (2004). Mining and Summarizing Customer Reviews. *KDD-2004 - Proc. Tenth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 168-177.

[17] V. B. and B. M. (2016). Analysis of Various Sentiment Classification Techniques. *Int. J. Comput. Appl.*, 140(3): 22-27.

[18] R. Joseph and G. Gray. (2020). Prediction of Student Academic Performance using Moodle Data from a Further Education Setting, 5(1).