Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/140215

Machine Learning for Improving Teaching Methods Through Sentiment Analysis

Thuraya Omran[1,2], *Baraa T. Sharef*[2,*], *Karim Hadjar*[3] *and Suresh Subramanian*[3]

[1]Department of Computer Science, College of Engineering, Design and Physical Sciences, Brunel University London, United Kingdom
[2]College of Information Technology, Department of Information Technology, Ahlia University, Manama, Kingdom of Bahrain
[3]College of Information Technology, Department of Multimedia Science, Ahlia University, Manama, Kingdom of Bahrain

Received: 3 Jan. 2020, Revised: 2 Feb. 2020, Accepted: 19 Feb. 2020
Published online: 1 Mar. 2020

Abstract: This paper describes how to use machine learning for improving teaching methods through collected sentiments from students. In fact, students sentiment analysis is a promising research area that is used to improve education by monitoring students performance and enabling students and lecturers to address teaching and learning issues in the most beneficial way. In our research, we aim to propose a machine-learning system for improving teaching methods through sentiment analysis, utilizing comments of students in reviews websites. The proposed system aims to automatically classify and analyze the students positive or negative feelings towards the current teaching process. Several techniques and procedures commonly used in natural language processing for the features processing task are used in designing and developing the proposed student sentiment analysis system. A total of 4000 comments of students were collected from RateMyProfessors.com website and used in the experiments of the current study. We have applied three supervised machine-learning techniques on these comments: Multinomial Naive Bayes (MNB), MaximumEntropy(MaxEnt), and Support Vector Machines (SVMs). The performance of the mentioned classifiers is evaluated using accuracy, precision, recall, and F1-score evaluation metrics.

Keywords: Natural Language Processing, Machine Learning Algorithm, Multinomial Naive Bayes, Logistic Regression, Support Vector Machines.

1 Introduction

Social media has tremendously changed the attitudes of most people. In fact, people were previously reluctant about expressing their opinions even when it is mandatory. However, thanks to the advent of smartphones and social media applications, the behavior of people has evolved and most of them are keen on expressing their opinions. These latter represent a rich source for the stakeholders and decision makers who have a deep interest in analyzing and evaluating the opinions or reviews. By nature, opinions are in unstructured form, they require Natural Language Processing (NLP) and machine learning algorithms in order to extract information and polarity from them [1]. In order to detect the attitudes and specify the polarity of these opinions, sentiment analysis can be used to achieve this purpose.

By applying sentiment analysis, sentiments can be detected at different levels, which could be at the document level, sentence level or aspect level [2]. Analysis of the document level and sentence level is close to subjectivity classification (whether it contains factual or subjective sentences) whereas analysis of aspect level focuses on the opinion itself.

To carry out these types of analysis, numerous approaches and algorithms are available which can be categorized into; Machine learning approach where classification based on labeled data, Lexicon-based approach and Hybrid approach. Each of these approaches has subcategories as presented in [3] and [4].

The current research aims to propose a machine learning system for improving teaching methods through sentiment analysis utilizing comments of students in reviews websites. It is also interesting in the document

* Corresponding author e-mail: baraa.nct@gmail.com

level of sentiments, whereas the concerned polarities are positive and negative.

In order to achieve these objectives, three experiments are carried out for several feature representation techniques, namely: Uni-gram feature representation, Bi-gram, and the combination of both of them.

This paper is organized as follows: Section 2 considers the related work; Section 3 presents the research methodology, while research results and discussions are discussed in section 4. Finally, Section 5 concludes the present research paper.

2 Related work

Natural Language Processing (NLP) is defined as a sub-field of Artificial Intelligence and computational linguistic and a significant component of text mining that enables the machine to understand the language of people [5]. Sentiment Analysis (SA), which is a field of NLP, is dedicated to studying the mechanism of utilizing machines for processing texts using language-processing algorithms [5]. It is also defined as a technique used in large data sources to specify opinions, feelings, appraisals toward objects and their attributes, services or product, whether this opinion was favorable or not [6]. Machine Learning (ML) is defined as a field of developing two types of algorithms: supervised and unsupervised for the purpose of clustering, classifying or predicting [7]. Supervised Machine Learning set of covariates (x) or features to predict the output (y), where there are observations with both x and y (training data), and the goal is to predict the (y) value in (test data) based on a given value of x [7].

The authors Esparza, et al. in 2016, deeply analysed the sentiment analysis steps. A proposed model for evaluating teacher performance presented in their study, on a dataset collected via twitter from a leading group of students from UPA institute of higher education [8]. Students comments are collected, cleaned and processed using suitable selection methods like Term Frequency-Inverse Document Frequency (TF-IDF). Support Vector Machines (SVM) is used on the dataset. In order to evaluate the classifier (SVM) performance, the authors used a certain mathematical Efficiency equation. One of the problems faced by the authors is in the corpus or datasets terms, terms were very general and were not focused on education. To overcome this problem, they added distinctive terms (features) to support the classification process.

A new model for measuring students sentiments towards e-learning materials has been proposed by Mandal, et al, in 2017 [9]. The collected dataset, which is fed into the processing and classifying engine, from a combination of polarity score and SVM have resulted in an accuracy of 85%. Moreover, the same dataset is fed into another classifier (Naive Bayes, MaxEntropy and intellimote) achieving good results based on four parameters like accuracy, recall, precision and time [9].

The researchers Rajput, et al, in 2016 carried out a study to analyze students textual feedback towards course delivery and instructor knowledge [10]. This feedback was collected as a step of course evaluation, in an automatic way and in order to develop qualitative and quantitative metrics, because such feedback was collected from a questionnaire containing open-ended questions and closed questions. The represented analysis for classifying the polarity of students' feedbacks to negative, positive or neutral, is done by generating sentiment score, word cloud for visualizing, and filters that are based on frequency. In order to classify the polarity of students feedback, the authors followed a procedure, which consists of the following steps: pre-processing stage, sentiment dictionary, polarity tagging, word frequency, word attitude, overall attitude, word cloud visualization, and sentiment score. The sentiment analysis is achieved by using Knime software, which is used to compute the sentiment score in order to classify the feedback polarity.

3 Research Methodology

Our research methodology for improving teaching methods through sentiment analysis using machine learning has a number of steps, which are described in the following sub-sections.

3.1 Collecting Data

Our dataset consists of 4000 comments extracted from the website Ratemyprofessors.com. The collected comments were distributed equally into two categories: positive and negative i.e. each category includes 2000 comments. The comments were saved into a *.csv* file and are manually labeled as positive or negative.

3.2 Pre-Processing Steps

The collected text comments have been processed according to the most common sub-processes: converting text to lowercase, stop word removal and tokenization, as illustrated in figure 1.

According to [11] the following steps clarify the main procedure proposed in the current research to pre-process the collected data: Step 1: Converting the letters in a text to lowercase, as an example letters 'A' will be converted to 'a'.
Step 2: Remove all the (stop words) which are the words that are not useful in sentiment analysis systems e.g. pronouns, prepositions, etc. The list of stop words used consists of (an, another, by, a, so, up, you, at, those, etc).
Step 3: Remove digits and punctuation marks for each

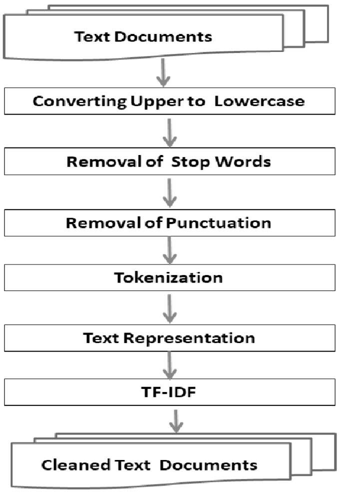


Fig. 1: Pre-processing steps

text in the dataset as example Remove (' * '. ' : ' , ' ?, ' \ ' etc).
Step 4: Split the text into tokens consisting of only words.

For example, if we have the following comment: "He is one of the best professors I have ever had, his lectures are fun". Then after performing the conversion of text to lowercase, removal of stop words and tokenization, the comment is as following

$$['best','professors','lectures','fun']$$

Step 5: The content of textual comments has been converted to feature terms which are composed of word strings that represent a suitable form for learning and categorization to be processed by the computer. These word strings that perform well in classification are extracted as feature terms in several previous studies. For this task unigram and bigram (Word-level) have been used as data representation methods in this study. They are effective as a language-independent method because they do not depend on the meaning of the language and work well in case of noisy text. In this research, a set of N extracted feature terms are expressed as follows:

$$T = \{tn|n = 1,2,?.,N\}. \tag{1}$$

N: total number of all features terms.

For unigram Word-level, when a word is used as a feature term. That is, each feature term tn corresponds to every single word. On the other hand, using the Bigram Word-level, each feature term tn corresponds to every two contiguous words.

Step 6: In this step, each feature is weighted by using Term Frequency-Inverse Document Frequency (TF-IDF) that reflects the importance of a word or feature to a document in a corpus or collection.

For more clarity, the idea is shown in the following example. Suppose that there are two types of documents (training and testing) as follows:

Training documents:
d1: the bag is heavy.
d2: the book is thick.
Test documents:
d3: the book in the bag is heavy.
d4: I can read the large book, the heavy book. The index vocabulary or the dictionary of the training documents d1 and d2 as following:

$$E(t)\begin{cases}1, \text{if t is "thick"}\\ 2, \text{if t is "book"}\\ 3, \text{if t is "heavy"}\\ 4, \text{if t is "bag"}\end{cases} \tag{2}$$

Since we have the dictionary of the training set, the test data could be converted to a vector space, by representing each term with index in the previous dictionary, i.e. the fourth term of the vector represents the word 'bag', where the first term of the vector represents the word 'thick' and so on.

To represent these terms of vector space that is indicated in $E(t)$ using Term Frequency (TF), it is nothing more than calculating the number of occurrences of each term in a test documents d3 and d4 using the formula of counting function:

$$tf(t,d) = \sum_{x\in d} fr(x,t) \tag{3}$$

where t denotes the term, d for a document, $fr(x,t)$ is a function defined as:

$$fr(x,t)\begin{cases}1, \text{if x = t}\\ 0, otherwise\end{cases} \tag{4}$$

which returns the number of term t occurrence in the d, for example, the $tf(book,d4) = 2$ and $tf(heavy,d3) = 1$. Since the vector notation of any document could be represented as:

$$v_{\vec{d_n}} = (tf(t_1,d_n),tf(t_2,d_n),tf(t_3,d_n),...tf(t_n,d_n)) \tag{5}$$

Then the vector representation of d3 and d4 can be as follows:

$$v_{\vec{d_3}} = (tf(t_1,d_3),tf(t_2,d_3),tf(t_3,d_3),...tf(t_n,d_3)) \tag{6}$$

$$v_{\vec{d_4}} = (tf(t_1,d_4),tf(t_2,d_4),tf(t_3,d_4),...tf(t_n,d_4)) \tag{7}$$

This results in the following vectors:

$$v_{\vec{d_3}} = \{0,1,1,1\} \tag{8}$$

$$\vec{v_{d_4}} = \{0,2,1,0\} \quad (9)$$

Since there is more than one document when they can be represented as a matrix in a form of $|D| \times F$, where D is the total number of documents and F is the number of features that are represented in vocabulary index as shown in the following equation:

$$M_{(|D| \times F)} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 2 & 1 & 0 \end{bmatrix} \quad (10)$$

As explained in [12,13], TF has a disadvantage in rising up the term that occurs frequently and vice versa. To overcome this point, TF-IDF is used to represent the term importance in a collection of documents as follows:

$$idf(t) = log((|D|)/(1+|d : t \in d|)) \quad (11)$$

Where $|d : t \in d|$ is the number of documents that contain the term t, ending up with $tf - idf$ equation as:

$$Tf - idf(t) = tf(t,d) \times idf(t) \quad (12)$$

After the preprocessing stage, the preprocessed data is divided in a random way into 10 subsets or folds with equal sizes using k-fold cross-validation procedure. Each time 10-1 subsets are put together as training set while one subset is used for testing, in order to evaluate the classifier performance later. A dictionary or index vocabulary for the training data and sparse matrix for test data are created.

The dictionary of training data enables the Term Frequency (TF) method to calculate the number of occurrences of each feature by representing it as vector space. Since TF scales down the rare features and scales up the redundant feature, another method like TF-IDF is used which reflects the importance of features in a whole collection of instances. TfidfVectorizer object in Python is used to calculate the TF-IDF. TfidfVectorizer contains analyzer, stop words and token pattern parameters which are responsible for achieving all these sub-processes, in addition to creating the dictionary and sparse matrix and representing the features in (N-gram) format using n-gram range parameter according to the desired one, weather it is a unigram, bigram, etc. In our research, the experiments have been performed three times using different n-gram (word-level) in order to compare the three classifiers performances in each experiment. In the first experiment, unigram (word-level) has been implemented followed by bigram (word-level) in the second experiment, while a combination of both unigram and bigram implemented in the third experiment. After finishing text preprocessing the cleaned data is fed to the three classification algorithms which are described in section 3.4.

The aim of these previous processes is to reduce the dimensionality of the dataset and to transform the comments from the plain text to a format which is suitable to the representation process and the other training and classification tasks.

3.3 Feature Extraction

In order to provide a short training time model with a reduction in dimensionality, the feature extraction stage is considered as a very critical one where a selection of distinctive attributes, features or variables are selected to build the model [8].

There are some methods of extracting features such as Linear Discriminant Analysis (LDA) and Latent Semantic Analysis (LSA)[14,15,16,17,18].In addition to countvectorizer which counts the number of feature occurrences in a document, while TF-IDF reflects the feature importance in the corpus (collected comments) using the following equation to calculate the value of each term [8].

$$wd = f_{w,d} * log(|D| f_{W,D}) \quad (13)$$

where D represents the collection of comments, w represents the term, d represents the individual comments owned by D, where $|D|$ denotes the corpus or dataset size, $f_{w,d}$ denotes the number of occurrences of w in d, while the number in which w occurs in D is represented by $f_{W,D}$.

The $tf - idf$ was calculated using TfidfVectorizer object in Python. TfidfVectorizer contains analyzer, stop words and token pattern parameters which are responsible for achieving all these sub-processes, in addition to creating the dictionary and sparse matrix and representing the features in (N-gram) format using n-gram range parameter according to the desired one, whether it is unigram, bigram, etc.

3.4 Training and Testing the Classification Methods

The classification process is based on supervised machine learning, as illustrated in figure 2, will be accomplished using three classifiers based on labeled and weighted features. The labeled features is converted to vectors using encoding which in turns calculates the features weight. In order to train and test the classifiers, the data is divided in a random way into 10 subsets or folds with equal size using k-fold cross-validation procedure. Each time 10-1 subsets are put together as training set while 1 subset is used for testing. This process can be done by importing cross-val-predict from sklearn.model-selection library involved in python. The classifiers used in the current research are:

1. Multinomial Naive Bayes (MNB) [19] where the document that is given to the (MNB) classifier is considered as words collection, that has class c with the probability $p(w|c)$ which observes a given word w in the corresponding class.

2. Maximum Entropy (MaxEnt) [20] where the feature conditional independence is not assumed.

3. Support Vector Machines (SVMs) [18] to determine a separator line with the best margin between different classes in search place.

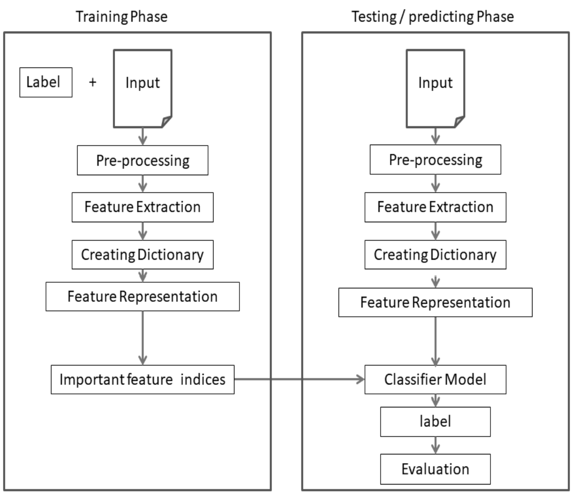


Fig. 2: Supervised machine learning phases

In supervised machine learning two stages are involved, training stage and testing stage, according to [11] in the training stage, data that are labeled under predefined categories are initially pre-processed in order to eliminate noisy and non-useful features. Next, features terms that become important keywords are extracted in the training stage from a representation process and getting important features indices. These indices are used later for the test stage. In the test stage, the investigated classifier is evaluated by classifying a set of pre-categorized data one by one as un-categorized data and then measuring the classification performance by using several standard techniques of performance evaluation.

3.5 Evaluating Parameters of Classifiers Performance

There are many ways to evaluate machine learning algorithms or classifiers performance. Some of these methods are ROC curves, measures function, and different methods of cross-validation [21], like hold out method-fold cross-validation, and leave-one-out cross-validation [22].

The intended method to be used in the current research is k-fold cross validation integrated with evaluation parameters like (Accuracy, Precision, Recall, and F1-score) to evaluate the performance of supervised machine learning algorithms.

K-fold cross-validation is a procedure used to evaluate the performance of the classifiers or to compare classifiers performance [23]. It is based on dividing the dataset to k-fold or k-subsets which are equal in size, each subset is used to test the model from k-1 folds. The classification performance of the classifier is evaluated by means of k accuracies that resulting from k-fold cross-validation [24]. The evaluation parameters (Accuracy, Precision, Recall, and F1-score) depend on a table called confusion matrix which includes labels of classes that are compared using terms like True Positive, True Negative, False Positive and False Negative [25,26]. Each measure is computed by sorting the classification result into the following:
True Positive (TP) and True Negative (TN) refers to the number of documents which are correctly assigned to the category. False Positive (FP) and False Negative (FN) refers to the number of documents which are falsely assigned to the category.
Accuracy is the ratio of documents that are correctly classified to the total number of documents.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{14}$$

Precision is the ratio of documents that are correctly labeled as positive to the total number of positively classified documents.

$$Precision = \frac{TP}{(TP+FP)} \tag{15}$$

Recall is the ratio of correctly-labeled values as positive to the total of a true positive & false negative.

$$Recall = \frac{TP}{(TP+FN)} \tag{16}$$

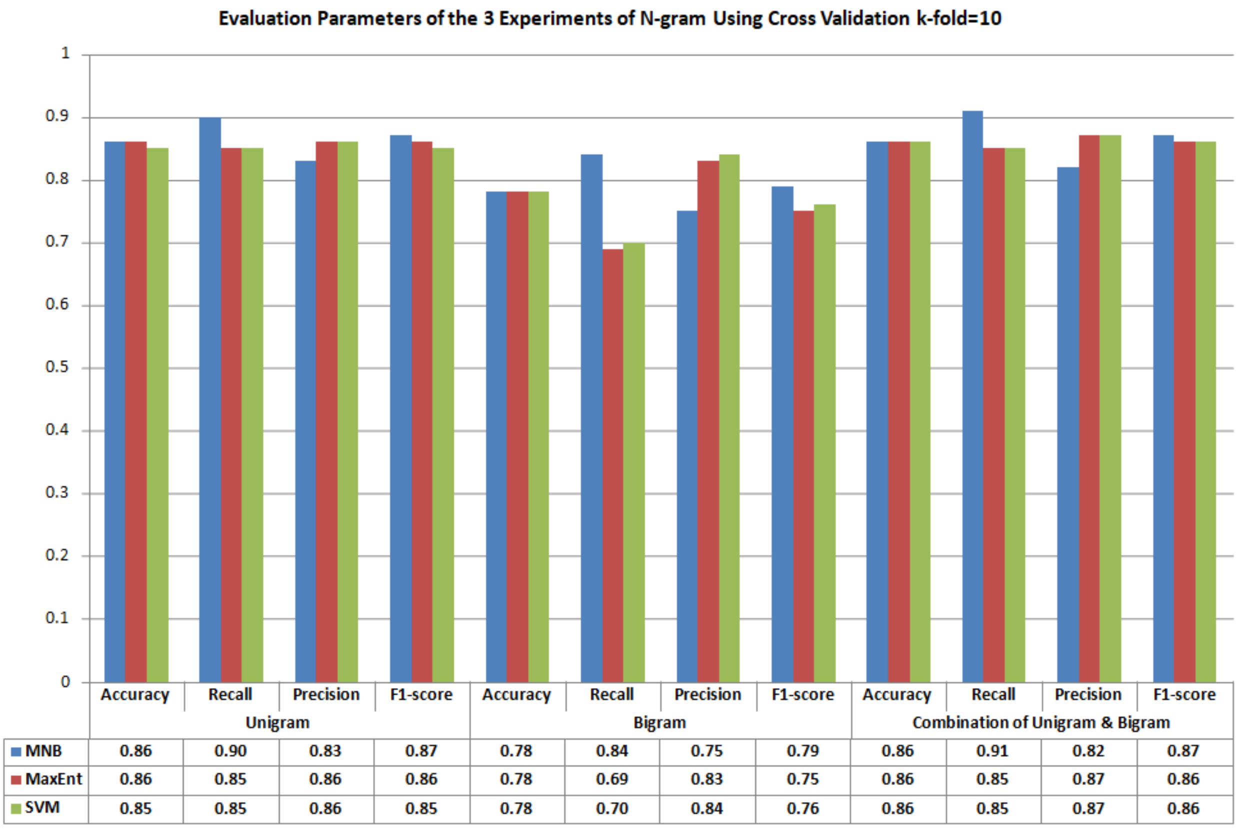
F-Measure or F1-score is the harmonic mean of recall and precision, which affect the final result of a system, by optimizing it toward one of them (recall or precision).

$$F-Measure = \frac{2*Precision*Recall}{Precision+Recall} \tag{17}$$

4 Results and Discussion

In the current research, three well-known machine learning algorithms are studied and examined on the collected data using different n-gram experiments and several proposed stages of sentiment analysis system are fixed for each experiment. The first experiment is performed using uni-gram feature representation; the second experiment is performed using bi-gram feature representation, while a combination of both unigram and bigram are performed in the third experiment. Different results of evaluation parameters are obtained for each classifier in each experiment.

Regarding the evaluation parameters of the first experiment as shown in figure 3, it can be observed that



	Unigram				Bigram				Combination of Unigram & Bigram			
	Accuracy	Recall	Precision	F1-score	Accuracy	Recall	Precision	F1-score	Accuracy	Recall	Precision	F1-score
MNB	0.86	0.90	0.83	0.87	0.78	0.84	0.75	0.79	0.86	0.91	0.82	0.87
MaxEnt	0.86	0.85	0.86	0.86	0.78	0.69	0.83	0.75	0.86	0.85	0.87	0.86
SVM	0.85	0.85	0.86	0.85	0.78	0.70	0.84	0.76	0.86	0.85	0.87	0.86

Fig. 3: Evaluation parameters of 3 experiments of n-gram using cross-validation k-fold=10

MaxEnt classifier has achieved 86%, 85%, 86% and 86% in terms of accuracy, recall, precision, and F1-score, respectively. Whereas, MNB classifier has achieved a performance of accuracy, recall, precision, and F1-score by 86%, 90%, 83% and 87%, respectively. On the other hand, SVM classifier has shown the lowest performance by 85%, 85%, 86% and 85% in terms of accuracy, recall, precision, and F1-score, respectively. The reason behind this is that the level of MaxEnt refers to its nature of non-assumption of feature conditional probabilities, in addition to the extraction of some features set from the input and combine them in a linear way and use of the sum in a form of exponent.

To estimate MaxEnt parameters in an accurate efficient way, we have considered some algorithms such as gradient ascent, conjugate gradient, methods of variable metric, and iterative scaling. Malouf, states that in order to accurately estimate the MaxEnt model, even the simplest model of MaxEnt, two factors are needed, a large quantity of annotated training data and considerable computational resources.

The estimation of MaxEnt model parameters for language processing is straightforward in concept, while practically it involves thousands of hundreds of free parameters which are expensive and sensitive in error rounding [16,7].

Regarding the evaluation parameters of the classifiers performance in the second experiment as shown in figure (3), it can be observed that SVM achieved 78%, 70%, 84% and 76% in regards to accuracy, recall, precision, and F1-score respectively, followed by MNB classifier which achieves performance of accuracy, recall, precision and F1-score by 78%, 84%, 75% and 79% respectively. On the other hand, MaxEnt classifier shows the performance achievement of 78%, 69%, 83%, and 75% respectively. The results behind the SVM achievement is because of its being linear non-probabilistic classifier that trains the model in order to find a hyper plan for separating data set by locating a best possible boundary that separates between negative and positive training documents.

It is also worth to mention, that there are parameters that affect SVM classifier performance such as the parameter of regularization, and gamma which represent tuning parameters of SVM. Varying the values of these parameters contributes to creating a nonlinear plane of classification with more accuracy in case of applications related to the real words of millions of training dataset that require finding the perfect class.

Another parameter is kernel which by a type of separation plan is defined to be linear or not. In addition to these parameters, there is a margin parameter which creates a separation line between classes points, knowing that the good margin is the one that fulfills a larger separation between classes.

Regarding the evaluation parameters of the classifiers in the third experiment as shown in figure 3, it can be said that MNB classifier achieves the highest performance of accuracy recall, precision and F1-score by 86%,91%,82% and 87% respectively, followed by MaxEnt which achieves performance of accuracy, recall, precision and F1-score by 86%,85%,87% and 86% respectively, in addition to the performance achievement by SVM of 86%,85%,87%,and 86% in regards to accuracy, recall, precision and F1-score respectively. The reason behind the best performance of MNB against all other classifier is due to the capability of MNB in updating the counts that are required to estimate the logarithmic and conditional probabilities in a straightforward way. A sentiment that is given to the MNB classifier is considered as words collection, that has a class with the probability which observes a given word in the corresponding class. This sentiment is estimated from the training data by computing the relative frequency for all words that are contained in the collection of training documents of that specific class.

Regarding the performance in each experiment, it is found that the parameter values in the combination of both unigram and bigram experiments are best when compared to the unigram experiment separately, as shown in figure 3. The reason behind such improvement in the performance is due to the rich features produced in such combination. These features affect the performance of the classification significantly.

5 Conclusionn

In the current research, well-known classification methods such as MNB, MaxEnt and SVMs are used to classify text of the collected comments of students based on sentiments appear in these comments

According to the obtained results, it is noticed that the MNB precision percentage is the lowest while its recall percentage is the highest in all experiments. One of the justifications for this level of performance is the fact that MNB has no tuning parameters. It can be said that the machine learning algorithm performance is affected by its input parameters because these parameters represent the algorithms settings. Knowing that the optimal or best setting of the learning algorithm depends on the scope of the problem that is at hand.

Some of the learning algorithms involved a large number of parameters that can be tuned in order to improve its performance. On the other hand, it is possible to say that these parameters have different effects on the classifiers performance domains and scopes, leading us to conclude that obtaining an optimal setting for learning algorithm is very hard or impossible.

As future work, we are planning to implement more classification algorithm using Arabic dataset, in addition to use more features selection methods and implementing a multi-class label classification process.

In addition, a graphical user interface could be designed that lets the user choose the classification method, with more sophisticated features. Furthermore, several feature selections and deep learning algorithms can be investigated to the huge amount of corpus to be collected from several resources such as social media and real-time systems.

References

[1] Tripathy, A., Agrawal, A. and Rath, S.K., Classification of sentiment reviews using n-gram machine learning approach, *Expert Systems with Applications.* **57**, 117-126 (2016).

[2] Devika, M.D., Sunitha, C. and Ganesh, A., Sentiment Analysis: A Comparative Study on Different Approaches. *Procedia Computer Science*, **87**, 44-49 (2016).

[3] Malouf, R., A comparison of algorithms for maximum entropy parameter estimation. *In proceedings of the 6th conference on Natural Language Learning*, Association for Computational Linguistic. 1-7, (2002).

[4] Zareapoor, M. and Seeja, K.R., Feature extraction or feature selection for text classification: A case study on phishing email detection. *International Journal of Information Engineering and Electronic Business*, 7(**2**), 60 (2015).

[5] Romero Llombart, O. and Duran Cals, J. *Using machine learning techniques for sentiment analysis*, Final Project On Computer Engineering, School Of Engineering(EE), Universitat Autonoma De Barcelona (UAB), Spain, (2017).

[6] Pawar, A.B., Jawale, M.A. and Kyatanavar, D.N., *Fundamentals of Sentiment Analysis: Concepts and Methodology*. Sentiment Analysis and Ontology Engineering, Springer International Publishing,Switzerland, 25-48, (2016).

[7] Mullainathan, S., and J. Spiess. Machine learning: an applied econometric approach, *Journal of Economic Perspectives*, 31(**2**) 87-106 (2017).

[8] Esparza, G.G., Diaz, A.P., Canul-Reich, J., De Luna, C.A. and Ponce, J. Proposal of a Sentiment Analysis Model in Tweets for improvement of the teaching-learning process in the classroom using a corpus of subjectivity. International *Journal of Combinatorial Optimization Problems and Informatics*, 7(**2**), 22 (2016).

[9] Mandal, L., Das, R., Bhattacharya, S. And Basu, P.N., Intellimote: a hybrid classifier for classifying learners' emotion in a distributed e-learning environment. *Turkish*

Journal of Electrical Engineering and Computer Sciences, 25(**3**), 2084-2095 (2017).
[10] Rajput, Q., Haider, S. and Ghani, S., Lexicon-Based Sentiment Analysis of Teachers Evaluation. *Applied Computational Intelligence and Soft Computing*, 1 (2016).
[11] Sharef, Baraa.T., Omar, N. and Sharef, Z.T., An automated arabic text categorization based on the frequency ratio accumulation. *Int. Arab J. Inf. Technol*, 11(**2**), 213-221 (2014).
[12] Mohammad, A.H., Alwadan, T. and Al-Momani, O., Arabic text categorization using support vector machine, Naive Bayes and neural network. *GSTF Journal on Computing (JoC)*, 5(**1**), 108 (2018).
[13] Cateni, S., Vannucci, M., Vannocci, M. and Colla, V., *Variable selection and feature extraction through artificial intelligence techniques*, Multivariate Analysis in Management, Engineering and the Sciences, 103-118 (2012).
[14] BRCK, G., Diri, B. and SONMEZ, A.C., Abstract feature extraction for text classification. *Turkish Journal of Electrical Engineering & Computer Sciences*, 20(**1**), 1137-1159 (2012).
[15] Do, H.H., Prasad, P.W.C., Maag, A. and Alsadoon, A., Deep learning for aspect-based sentiment analysis: a comparative review. *Expert Systems With Applications*, **118**, 272-299 (2019).
[16] Xie, X., Ge, S., Hu, F., Xie, M. and Jiang, N., An improved algorithm for sentiment analysis based on maximum entropy. *Soft Computing*, 23(**2**), 599-611 (2019).
[17] Yasen, Khaled N., Fahad Layth Malallah, Lway Faisal Abdulrazak, Aso Mohammad Darwesh, Asem Khmag, and Baraa T. Sharef. Hand detection and segmentation using smart path tracking fingers as features and expert system classifier. *International Journal of Electrical and Computer Engineering (IJECE)* 9(**6**), 5277-5285 (2019).
[18] Ahmad, M., Aftab, S., Muhammad, S.S. and Ahmad, S., Machine Learning Techniques for Sentiment Analysis: A Review. *Int. J. Multidiscip. Sci. Eng*, 8(**3**), 27-32 (2017).
[19] Xu, S., Li, Y. and Wang, Z., *Bayesian Multinomial Naive Bayes Classifier to Text Classification*. Advanced multimedia and ubiquitous engineering. Springer, Singapore. 347-352, (2017).
[20] Xie, X., Ge, S., Hu, F., Xie, M. and Jiang, N., An improved algorithm for sentiment analysis based on maximum entropy. *Soft Computing*, 23(**2**), 1-13 (2017).
[21] Lavesson, N., *Evaluation of classifier performance and the impact of learning algorithm parameters*. M.A thesis, Blekinge Institute of Technology, Swedish, (2003).
[22] Cs.cmu.edu. (2018). Cross Validation. [online] Available at: https://www.cs.cmu.edu/ schneide/tut5/node42.html [Accessed 25 October 2019].
[23] M. Zidan, A.-H. Abdel-Aty, A. El-Sadek, E. A. Zanaty, and M. Abdel-Aty, *Low-cost autonomous perceptron neural network inspired by quantum computation*, AIP Conference Proceedings, 020005, 2017.
[24] Wong, T.T, Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition*, 48(**9**), 2839-2846 (2015).
[25] M. Zidan, A. H. Abdel-Aty, M. El-shafei , M. Feraig, Y. Al-Sbou, Hichem Eleuch and M. Abdel-Aty . Quantum Classification Algorithm Based on Competitive Learning Neural Network and Entanglement Measure. *Appl. Sci.*, 9(**7**), 1277 (2019).
[26] Ting, K.M., *Confusion matrix. In Encyclopedia of Machine Learning and Data Mining*. Springer Publishing Company, Incorporated, Boston, 260-260 (2017).

Thuraya Omran awarded her Master's degree in information technology and computer science from Ahlia university-Bahrain in 2018. Her research interest in the area of text mining, natural languages processing and machine learning. She recently applied for PhD in information systems and computing research at Brunel university London.

Baraa T. Sharef is a Ph.D. holder in Information Science and Technology from National University of Malaysia, since 2016. His BSc and MSc are both in IT. Currently, he is an Assistant Professor of Information Technology at Ahlia University-Bahrain. He has been appointed as international advisory reviewer in several international journals and conferences indexed in Scopus and ISI.He handles students from diverse nationalities and backgrounds. He has contributed in teaching the hosted master level students from Epitech-France. Moreover, he has been appointed as recognized supervisor for PhD students in Brunel University London-UK. His interest in research are rich in many areas such as, Information retrieval, Natural language processing, Intelligent systems, Machine learning, Multi-Criteria Decision Making, Wireless network operation and structure and Knowledge discovery.

-

Karim Hadjar Received the PhD degree in Computer Science at University of Fribourg in Switzerland. His research interests are in the areas of Document Recognition: Physical and Logical Layout Analysis, Machine Learning and Pattern Recognition, Image Processing, Security and Privacy, Big Data, e-Learning and Web Services Ontologies. He has published research articles in reputed international journals and reputed conferences.

Suresh Subramanian is Assistant Professor of Computer Science at Ahlia University Kingdom of Bahrain. His research interests are in the areas of Machine learning including the area of sentiment analysis, email security and deep learning techniques for analyzing the English and Arabic text. Published research papers in the renowned journals and the peer reviewer for international journals. Professional software developer using Microsoft technology such as .Net and Sql Server. Microsoft Certified Professional Developer (MCPD) in ASP .Net and Sharepoint and recently completed the Amazon Web Services Certified Cloud Practitioner certification. Ardent toastmaster and social worker.