ISSN 2278-3091
Volume 9, No.3, May - June 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse216932020.pdf
https://doi.org/10.30534/ijatcse/2020/216932020

Multi-class Document Classification using Support Vector Machine (SVM) Based on Improved Naïve Bayes Vectorization Technique

Hajah T. Sueno[1], Bobby D. Gerardo[2], Ruji P. Medina[3]
[1]Technological Institute of the Philippines-Quezon City, Philippines, qhsueno@tip.edu.ph
[2]West Visayas State University, Philippines, bgerardo@wvsu.edu.ph
[3]Technological Institute of the Philippines-Quezon City, Philippines, ruji.medina@tip.edu.ph

ABSTRACT

At present several vectorization approaches are used to transform text documents into a numerical format. A huge number of features converted from text data from a single document take time to process vectorized data with large dimensions. To reduce the number of dimensions, this work uses an improved Naïve Bayes algorithm to vectorize documents according to a distribution of probabilities reflecting the probable categories to which the document that belongs. The improved Naïve Bayes vectorization used Laplace smoothing to ensure that posterior probabilities are never zero and logarithmic function to solve the result of the probability calculation that is too small that cannot be represented. The text classification algorithms based on the vector space model, such as the Support Vector Machine (SVM), use this probability distribution as the vectors to represent the document that is used to classify the documents. To validate the improvement of the Naïve Bayes vectorization technique, the results are compared to TF-IDF vectorization. The results showed that the transformation of data by improved Naïve Bayes vectorization technique reduces dimensionality and has contributed to better performance of the SVM classification approach.

Key words: Document classification, Naïve Bayes Vectorization, Support Vector Machine, Vectorization

1. INTRODUCTION

Classification of documents can be characterized as the task of categorizing collections of electronic documents automatically into their annotated classes, based on their content[1]. For text documents to be used in text classification, it needs to be processed and transformed from the text version to a document vector, making it much easier to manage and reduce the dimensionality of features [2]. Features in machine learning are numerical attributes from which anyone can perform some mathematical operation. But there are various situations when the dataset does not contain numerical attributes. These types of text data cannot directly be fed in the machine for extracting features, as most of the algorithms expect the feature vectors of the text as input [3]. The words of text documents are usually explicitly vectorized, turning the text documents into a numerical format. The significant number of features transformed from the text data of a document makes the classifiers take time to process large vectorized data. Support vector machines (SVM) can be used as a discriminative classifier of documents and have proved to be more accurate than most other classification techniques [4][5]. To improve the generalization of the overall system, this study introduces an improved Naïve Bayes vectorization technique with a smoothing technique to overcome zero probability of unseen data and application of the logarithmic function to avoid underflow error. To reduce the number of features, this study introduces an improved vectorization technique using Naive Bayes as the vectorizer for the text documents by using the probability distribution, where the dimension of the features is based on the number of available categories in the classification task. The technique takes advantage of the simplicity of Naïve Bayes and the accuracy of SVM.

2. LITERATURE REVIEW

2.1 Dimensionality Reduction

Reduction of dimensionality is usually used to reduce a large collection of data to its most discriminative components to provide specific information and to define it with fewer features[6][7]that will automatically increase the performance of the classifier by decreasing the execution time and space complexity [8]. It is carried out before classification so that the classifiers can be constructed in a simple way to measure. By doing so, however, it must also be reliable and must not lead to information loss. An effective classification must also be carried out using suitable methods for the reduction of measurements [9]. The reduction of dimensionality increases the efficiency of the F-score analysis classification problem. F-score, on the other hand, is an easy and efficient technique for selecting meaningful information from the high dimensional data[10]. The F1-score results obtained from previous test scenarios showed that classification using the dimension reduction process performed better in selecting features in [11].

2.2 Naïve Bayes Vectorization

Machine learning algorithms most commonly take numeric feature vectors as input for automatic learning, extraction and analysis[12]. Thus, when working with text documents, a way to convert each document into a numeric vector is needed. This process is known as text vectorization. This technique aims to build a new set of features after applying a few transformations into the corpus. It enables the machines to understand the textual contents by converting them into meaningful numerical representations [13]. Usually, the extracted features do not carry the same information as before because of the transformations. However, it is possible to achieve a more condensed data providing a great dimension reduction[14].Vector space representation of text is used in various text classification problems. The key idea is that the context of words in a particular document is captured and has a better representation, hence it can help build a better classifier [15].

Naive Bayes vectorization for text documents used the probability distribution, where the dimension of the features is based on the number of available categories in the classification task [16]. It uses the raw text document for training purposes, and the classifier uses the vectorized training data supplied by the vectorizer [17]. Naive Bayes vectorization focuses on the Bayes formula with presumed independence among predictors, using a set of training data to calculate the posterior probability, which is calculating the likelihood and estimates the probability terms needed for classification[18][19]. In the context of document classification, the Naive Bayes vectorization uses the probability of a particular document being annotated to a particular category, given that the document contains certain words in it, is equal to the probability of finding those particular words in that category, times the probability that any document is annotated to that category, divided by the probability of finding those words in any document[20][21], as shown in equation (1):

$$Pr\ (Category|Word) = \frac{Pr\ (Word|Category) * Pr\ (Category)}{Pr\ (Word)} \qquad (1)$$

Each document contains words which are given probability values based on the number of its occurrence in the document. Many researchers proved the effectiveness of using Bayes theorem in various domains, such as in health [22][23], agriculture [24][25], and image processing [26][27]. Meneses et al. [28] have presented two simple approaches to approximate Bayes formula while making accurate decisions. The performance was assessed where a decision is made on which of two occurrences is most likely to occur and where a choice is made between an option that offers acceptable usefulness for something that is certain or for a risk that results in either a worse or better value. Bayes theorem was also studied in classifying thousands of Navigational Talexmessages gathered in navigational area VI for an effective and safe intelligent navigation system. Based on the result, the accuracy rate of the optimal classifier reaches 97 percent[29]. One closely related research paper to this study was of [30][31], that have used a Naive Bayes vectorization technique to preprocess the text documents and decrease the dimensionality. The results also improved the accuracy of the classification.

2.3 Smoothing Method

Smoothing is a method which adjusts the maximum likelihood estimate to correct a non-zero probability to unseen words and increases the accuracy of the model due to data sparseness [32]. The general form of the smoothed model is of based on equation (2):

$$P(w|d) = \begin{cases} P_s\ (w \mid d) & \text{if w is seen} \\ \alpha d\ P(w|C) & \text{otherwise} \end{cases} \qquad (2)$$

Where P_s (w | d) is the smoothed probability word seen in the document and P(w|C) is the group language model and αd is the coefficient controlling the probability assigned to unseen words so that probabilities sum to one. In general, smoothing methods differ in the choice of Ps (w | C). It can be as simple as adding extra count or more complex where words of the different count are treated differently. Not only do smoothing methods generally prevent zero probabilities, but they also attempt to improve the accuracy of the model as a whole [33][34].

Chen and Goodman [33] previously examined smoothing methods for language modeling, involving additive smoothing (Laplace/Lidstone), Good-Turing, Jelinek-Mercer, Katz smoothing, Witten-Bell, Absolute Discounting, and KneserNey. For comparison evaluation, they used the measure cross-entropy for evaluating language models. The best smoothing method according to this study is their modification of Kneser-Ney smoothing.Researches on smoothing methods were further studied in text classification with Naive Bayes. For spam email classification, Hafilizara[35] compared Laplace, JelinekMercer, Dirichlet, Absolute Discounting, and Two-Stage smoothing. The results revealed that the Dirichlet smoothing method provided the best performance. For question topic classification, Yuan et al. [36] studied four smoothing methods for Naive Bayes: Jelinek-Mercer, Dirichlet, Absolute Discounting, and Two-Stage (TS). Their result showed that Absolute Discounting and TS are the two best-performing methods.

Smoothing methods have also been studied in conducting real-time stream data which investigated a scheme adopting

Laplace smoothing technique with Binarized Naïve Bayes Classifier (NBC) for enhancing the accuracy, and employing SparkR for speed up.[37].

2.3 Support Vector Machine (SVM)

SVM is a non-probabilistic linear binary classifier, supervised models of learning described by a separating hyperplane. It uses a separating hyperplane or a decision plane to demarcate decision boundaries among a set of data points classified with different labels. It is a strictly supervised classification algorithm. In other words, the algorithm develops an optimal hyperplane utilizing input data or training data, and this decision plane in turn categories new examples [6]. In other words, provided the labeled training data, the algorithm produces an optimal hyperplane which classifies new examples. The operation of the SVM algorithm is based on finding the hyperplane which gives the training examples the largest minimum distance. Within the SVM theory, this distance is called a margin [38], [39]. A hyperplane is constructed in this feature space that maximizes the separation margin between the hyperplane and the points located closest to it as a supporting vector [39]. The best hyperplane is the one which is the biggest margin. If such a hyperplane exists, it is known as the ―maximum margin hyperplane, and the linear classifier it defines is known as a maximum margin classifier or the perceptron of optimal stability [40].

3. METHODOLOGY

In this study, an improved Naive Bayes vectorization model is developed to reduce the dimensionality of data and to produce a higher classification accuracy. The proposed improved Naïve Bayes vectorization as discuss in our previous study [41] was done by applying Laplace smoothing and utilizing Logarithmic function. The steps to implement and validate the performance of the improved vectorization include data and preprocessing, vectorization, and evaluation is shown in Figure 1.

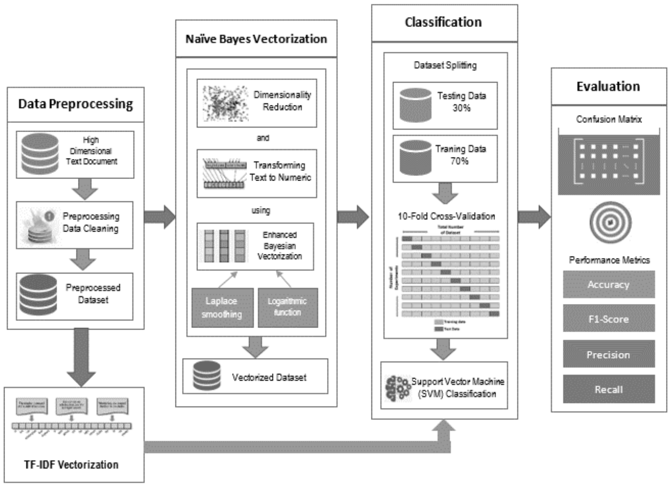


Figure 1: Improved Naïve Bayes Vectorization Model

3.1 Data Source

The proposed improved Naïve Bayes vectorization has been tested and evaluated using three datasets.

The WebKB dataset which had been utilized in the study of was acquired from Ana Cardoso-Cachopo's[42] website consists of four categories - Faculty, Student, Course, and Project with a total of 4199 documents. The training set is constructed by 2803 documents, while the testing set consists of 1396 documents.

The song lyrics are gathered from different websites such as Lyrics (www.lyrics.com), Genius (www.genius.com), and Musixmatch (www.musixmatch.com) with a total of 325 song lyrics. The determination of category based on five categories – Love songs, Christmas songs, Friendship songs, Worship songs, and Nationalism songs are manually annotated to determine the labels to be used for supervised learning.

News headlines from the year 2012 to 2018 obtained from HuffPost which was originally collected by Misra[43]. Using the data tool of a spreadsheet application, each news headline was manually labeled according to the following five categories namely – Urban, Science, Arts, Politics, and Travel news with a total of 2640 documents.

The study uses the same training dataset for both the Naive Bayes vectorizer and the SVM classifier. The Naive Bayes vectorizer uses the labeled text document as training data, and the classifier uses the vectorized training data supplied by the Naive Bayes vectorizer.

3.2 Preprocessing Methods

In this stage, terms standardization is done to eliminate accentuation, punctuation, special characters, and numbers. All letters are converted into lowercase letters, every character except alphabets and multiple spaces are replaced by single space. Noisy data such as text file header, footer, HTML, XML, and markup data are removed.

3.3 Building the Improved Naïve Bayes Vectorization

The initial step of analyzing the text document is by grouping each document in the training set by its category. A simple word extraction algorithm is used to extract each word from the document (X) to generate a list of words containing the number of occurrences of each word in the category (C). The same method is used to generate the sum of all words in every category in the training data set.

The prior probability of every category can then be computed using equation (3):

$$Pr\,(C) = \frac{\underline{\text{Total Number of Document in Category}}}{\text{Total\_Number\_Of\_Document\_in\_Training\_Dataset}} \qquad (3)$$

To calculate the likelihood of a particular category for a particular word, the equation (4) below is used.

$$Pr\ (X|\ C) = \frac{\text{Occurrence of Word In the Category}}{\text{Total\_Number\_of\_All\_Words\_in\_Category}} \tag{4}$$

When you have a model with many features, the entire probability will become zero because one of the feature's value was zero. Laplace smoothing is applied to avoid zero probability situation and to ensure that each word has a probability of occurrence, based on at least a single count, even if it does not appear in the training data. The count is increased to a small value (usually 1) [32][44] using the equation (5):

$$Pr\ (X|\ C) = \frac{\text{Occurrence of Word In the Category } +1}{\text{Total\_Number\_of\_All\_Words\_in\_Category} + |V| +1} \tag{5}$$

where |V| is the total unique words in the training set

Based on the derived Bayes' formula for text classification, *Pr(Category)* is prior probability,*Pr(Word|Category)* is the likelihood, and *Pr(Word)* is the evidence, the posterior probability Pr*(Category | Word)* of each word (W) in the input document annotated to each category can be measured. The overall probability for a document to be annotated to a particular category is calculated using the equation (6):

$$Pr\ (C|\ X) = Pr\ (W_1|\ C) * Pr\ (W_2|C) * Pr\ (W_3|C) * Pr\ (W_n|C) * Pr\ (C) \tag{6}$$

Since the numerical values of probabilities of words are relatively small, multiplying all these probabilities to find the product will produce a smaller numerical value that frequently results in underflow which means that for that given test sentence, the trained model will fail to predict the category. To avoid this underflow error, a mathematical *log* is applied using the equation (7):

$$Pr\ (C\ |\ X) = log\ (Pr\ (W_1|C) + log\ (Pr\ (W_2|C) + log\ (Pr\ (W_3|C) + log\ (Pr\ (W_n|C) + log\ (Pr\ (C)) \tag{7}$$

The logarithmic function is applied since *log* increases or decreases monotonically which means that it will not affect the order of probabilities. Smaller probabilities will still stay smaller after the log has been applied to them and vice versa. For example, if the test word "very" has a smaller probability than the test word "happy", so after passing these through log would although increase their magnitude but "very" would still have a smaller probability than "happy". Therefore, without affecting the predictions of the trained model, it can effectively avoid the common pitfall of underflow error [45].

The right category is characterized by the category that has the highest posterior probability value, Pr(C|X), as stated in the Bayes Classification Rule [19].

3.4 Evaluating the Model

After creating the improved model, the model was trained using a set of well-categorized vectorized training data supplied by the Naive Bayes vectorizer. LIBSVM[46]machine learning toolkit was used to train the classification models. The model was trained using a set of well-categorized vectorized training data supplied by the Naïve Bayes vectorizer. The vectorized training data was split into a 70% training set and a 30% testing set or classification by performing the SVM. To evaluate the robustness of the estimates from the SVM models, 10-fold cross-validation was performed in the training data set. The rest of the classification tasks is performed using the linear kernel function with the implementation of parameter C that is set to 1.

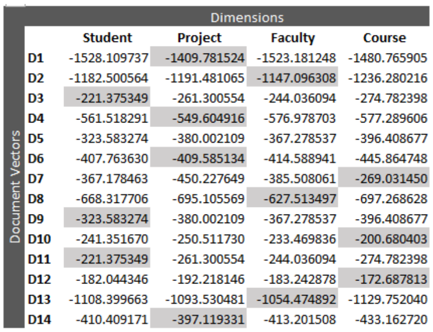
The enhanced Bayesian vectorization was evaluated using precision, recall, F1-score, and accuracy as the performance measures and the confusion matrix to calculate the classification accuracy. By comparing the results by SVM, the following quantities are calculated: true positive (TP) is the number of correctly classified as positive, false negative (FN) is the number of positive that is incorrectly classified as negative, true negative (TN) stands for the number of correctly classified as negative, and false positive (FP) refers to the number of negative incorrectly classified as positive.

4. SIMULATION RESULTS AND DISCUSSIONS

3.4 Evaluating the Model

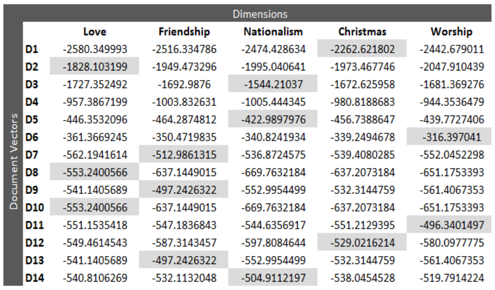
Table 2, Table 3, and Table 4 show the vectorized datasets. The SVM classifier uses the vectorized training data supplied by the Naïve Bayes vectorizer using the probability of distribution, thus the dimension of the features is based on the number of available categories in the classification task. The values in the document vector represent the document's distributed weight across dimensions. In a simplified sense, each dimension represents a meaning and the document's numerical weight on that dimension captures the closeness of its association with and to that meaning. The highlighted value in a document represents the highest probability which implies the dimension or category it belongs.

Table 2: WebKB Vectorized Dataset



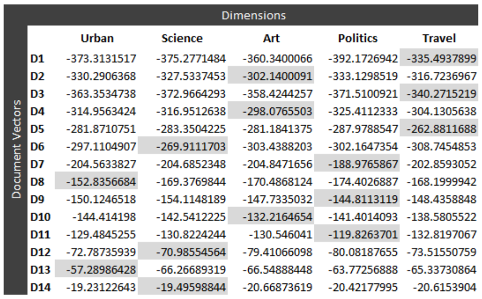
Document Vectors		Dimensions			
		Student	Project	Faculty	Course
	D1	-1528.109737	-1409.781524	-1523.181248	-1480.765905
	D2	-1182.500564	-1191.481065	-1147.096308	-1236.280216
	D3	-221.375349	-261.300554	-244.036094	-274.782398
	D4	-561.518291	-549.604916	-576.978703	-577.289606
	D5	-323.583274	-380.002109	-367.278537	-396.408677
	D6	-407.763630	-409.585134	-414.588941	-445.864748
	D7	-367.178463	-450.227649	-385.508061	-269.031450
	D8	-668.317706	-695.105569	-627.513497	-697.268628
	D9	-323.583274	-380.002109	-367.278537	-396.408677
	D10	-241.351670	-250.511730	-233.469836	-200.680403
	D11	-221.375349	-261.300554	-244.036094	-274.782398
	D12	-182.044346	-192.218146	-183.242878	-172.687813
	D13	-1108.399663	-1093.530481	-1054.474892	-1129.752040
	D14	-410.409171	-397.119331	-413.201508	-433.162720

Table 3: Song Lyrics Vectorized Dataset



		Dimensions				
		Love	Friendship	Nationalism	Christmas	Worship
Document Vectors	D1	-2580.349993	-2516.334786	-2474.428634	-2262.621802	-2442.679011
	D2	-1828.103199	-1949.473296	-1995.040641	-1973.467746	-2047.910439
	D3	-1727.352492	-1692.9876	-1544.21037	-1672.625958	-1681.369276
	D4	-957.3867199	-1003.832631	-1005.444345	-980.8188683	-944.3536479
	D5	-446.3532096	-464.2874812	-422.9897976	-456.7388647	-439.7727406
	D6	-361.3669245	-350.4719835	-340.8241934	-339.2494678	-316.397041
	D7	-562.1941614	-512.9861315	-536.8724575	-539.4080285	-552.0452298
	D8	-553.2400566	-637.1449015	-669.7632184	-637.2073184	-651.1753393
	D9	-541.1405689	-497.2426322	-552.9954499	-532.3144759	-561.4067353
	D10	-553.2400566	-637.1449015	-669.7632184	-637.2073184	-651.1753393
	D11	-551.1535418	-547.1836843	-544.6356917	-551.2129395	-496.3401497
	D12	-549.4614543	-587.3143457	-597.8084644	-529.0216214	-580.0977775
	D13	-541.1405689	-497.2426322	-552.9954499	-532.3144759	-561.4067353
	D14	-540.8106269	-532.1132048	-504.9112197	-538.0454528	-519.7914224

Table 4: News Headlines Vectorized Dataset

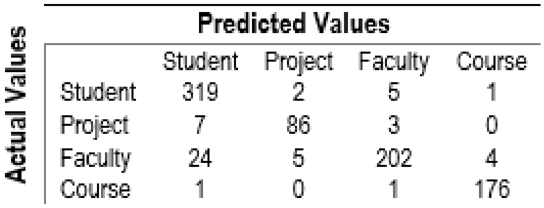


		Dimensions				
		Urban	Science	Art	Politics	Travel
Document Vectors	D1	-373.3131517	-375.2771484	-360.3400066	-392.1726942	-335.4937899
	D2	-330.2906368	-327.5337453	-302.1400091	-333.1298519	-316.7236967
	D3	-363.3534738	-372.9664293	-358.4244257	-371.5100921	-340.2715219
	D4	-314.9563424	-316.9512638	-298.0765503	-325.4112333	-304.1305638
	D5	-281.8710751	-283.3504225	-281.1841375	-287.9788547	-262.8811688
	D6	-297.1104907	-269.9111703	-303.4388203	-302.1647354	-308.7454853
	D7	-204.5633827	-204.6852348	-204.8471656	-188.9765867	-202.8593052
	D8	-152.8356684	-169.3769844	-170.4868124	-174.4026887	-168.1999942
	D9	-150.1246518	-154.1148189	-147.7335032	-144.8113119	-148.4358848
	D10	-144.414198	-142.5412225	-132.2164654	-141.4014093	-138.5805522
	D11	-129.4845255	-130.8224244	-130.546041	-119.8263701	-132.8197067
	D12	-72.78735939	-70.98554564	-79.41066098	-80.08187655	-73.51550759
	D13	-57.28986428	-66.26689319	-66.54888448	-63.77256888	-65.33730864
	D14	-19.23122643	-19.49598844	-20.66873619	-20.42177995	-20.6153904

3.4 Classification Results

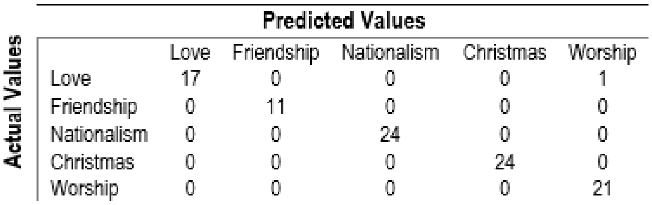
Table 5, Table 6, and Table 7 show correctly classified number of observations. In Table 5, the SVM classifier correctly classified 318 Students, 86 Projects, 202 Faculty, and 176 Courses. In Table 6, the SVM classifier correctly classified 17 love songs, 11 friendship songs, 24 nationalism songs, 24 Christmas songs, and 21 worship songs. In Table 7, the SVM classifier correctly classified 180 urban news, 173 science news, 118 art news, 114 politics news, and 181 travel news.

Table 5: Confusion Matrix for WebKB Dataset



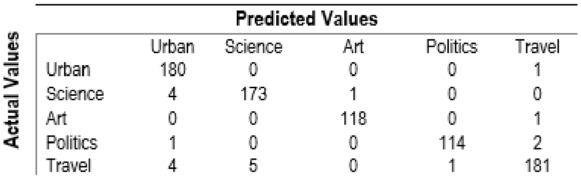
		Predicted Values			
		Student	Project	Faculty	Course
Actual Values	Student	319	2	5	1
	Project	7	86	3	0
	Faculty	24	5	202	4
	Course	1	0	1	176

Table 6: Confusion Matrix of Song Lyrics Dataset



		Predicted Values				
		Love	Friendship	Nationalism	Christmas	Worship
Actual Values	Love	17	0	0	0	1
	Friendship	0	11	0	0	0
	Nationalism	0	0	24	0	0
	Christmas	0	0	0	24	0
	Worship	0	0	0	0	21

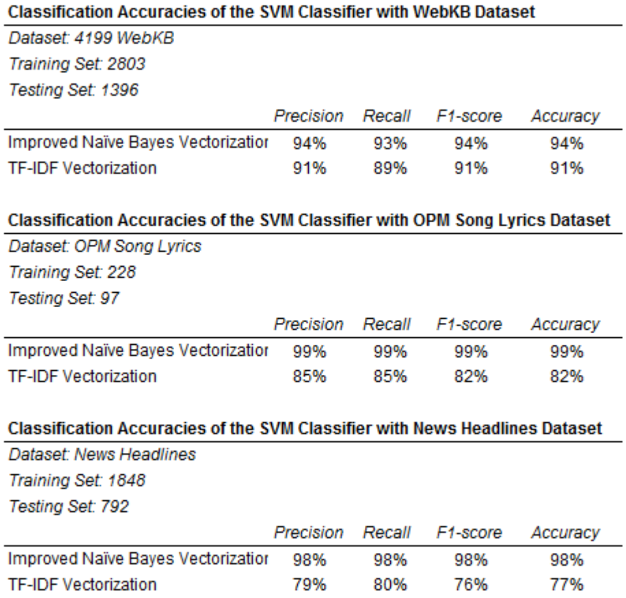
Table 7: Confusion Matrix of News Headlines Dataset



		Predicted Values				
		Urban	Science	Art	Politics	Travel
Actual Values	Urban	180	0	0	0	1
	Science	4	173	1	0	0
	Art	0	0	118	0	1
	Politics	1	0	0	114	2
	Travel	4	5	0	1	181

The Naïve Bayes vectorization technique had been utilized to transform textual data into a numerical format. On the other hand, the TFIDF (Term Frequency Inverse Document Frequency) technique has been reported as one of the most widely used pre-processing techniques by many text mining research groups for the same purpose. To validate the improvement of using Naïve Bayes as a vectorization technique, the study compared the classification performance over the TF-IDF vectorization for the SVM classifier. As shown in Table 8, the improved vectorization technique model achieved a significantly higher Accuracy than the classification method of using TF-IDF. The improved technique also yields the highest values for Precision, Recall, and F1-score.

Table 8 : Comparison of using Improved Naïve Bayes Vectorization and TF-IDF Vectorization in Classification



Classification Accuracies of the SVM Classifier with WebKB Dataset

Dataset: 4199 WebKB
Training Set: 2803
Testing Set: 1396

	Precision	*Recall*	*F1-score*	*Accuracy*
Improved Naïve Bayes Vectorizatior	94%	93%	94%	94%
TF-IDF Vectorization	91%	89%	91%	91%

Classification Accuracies of the SVM Classifier with OPM Song Lyrics Dataset

Dataset: OPM Song Lyrics
Training Set: 228
Testing Set: 97

	Precision	*Recall*	*F1-score*	*Accuracy*
Improved Naïve Bayes Vectorizatior	99%	99%	99%	99%
TF-IDF Vectorization	85%	85%	82%	82%

Classification Accuracies of the SVM Classifier with News Headlines Dataset

Dataset: News Headlines
Training Set: 1848
Testing Set: 792

	Precision	*Recall*	*F1-score*	*Accuracy*
Improved Naïve Bayes Vectorizatior	98%	98%	98%	98%
TF-IDF Vectorization	79%	80%	76%	77%

Furthermore, by applying the improved Naïve Bayes vectorization technique to preprocess the documents, the textual data is transformed into a numerical format, thereby reducing the dimensionality of the data resulting in higher F1-scores.

5. CONCLUSION

Based on the comparison using the three datasets the improved Naive Bayes-SVM classifier outperforms the TF-IDF. It has been observed that the proposed improvement is highly efficient and classifies the documents with great accuracy. The employment of Laplace smoothing to the enhancement of Naive Bayes-SVM has achieved a classification a higher accuracy compared to the TF-IDF. These results showed that the Naive Bayes vectorization technique has contributed a more effective textual data transformation process to the SVM classifier, as compared to the use of the TFIDF vectorization technique for the same purpose.

Future directions of the research will be the exploration of other features and weights to produce word vectors and investigates the effect of other smoothing methods to Naive Bayes vectorization.

ACKNOWLEDGEMENT

This work was supported by the Commission of Higher Education K-12 Scholarship Program. We also thank the Technological Institute of the Philippines, Quezon City (TIP-QC) for providing us with helpful feedback and suggestions.

REFERENCES

1. K. Roy, "Classification of Text Documents Through Multi-Domain Bangla Text Documents," 2017.
2. M. K. Elhadad, K. Badran, and G. I. Salama, "A novel approach for ontology-based dimensionality reduction for web text document classification," *Proc. - 16th IEEE/ACIS Int. Conf. Comput. Inf. Sci. ICIS 2017*, pp. 373–378, 2017, doi: 10.1109/ICIS.2017.7960021.
3. D. Saxena, S. K., and K. N., "Survey Paper on Feature Extraction Methods in Text Categorization," *Int. J. Comput. Appl.*, vol. 166, no. 11, pp. 11–17, 2017, doi: 10.5120/ijca2017914145.
4. A. H. Aliwy and E. H. A. Ameer, "Comparative Study of Five Text Classification Algorithms with their Improvements," *Int. J. Appl. Eng. Res. ISSN*, vol. 12, no. 14, pp. 973–4562, 2017, [Online]. Available: http://www.ripublication.com.
5. M. H. Bhavsar and D. A. Ganatra, "Support Vector Machine Classification using Mahalanobis Distance Function," *Int. J. Sci. Eng. Res.*, vol. 6, no. 1, pp. 618–626, 2015, doi: 10.14299/ijser.2015.01.006.
6. K. Das and R. N. Behera, "A Survey on Machine Learning: Concept, Algorithms and Applications," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 5, no. 3, pp. 1301–1309, 2017, doi: 10.15680/IJIRCCE.2017.
7. V. R. Sayoc, T. K. Dolores, M. C. Lim, L. Sophia, and S. Miguel, "Nature Inspired Dimensional Reduction Technique for Fast and Invariant Visual Feature Extraction," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 3, pp. 195–200, 2019, doi: https://doi.org/10.30534/ijatcse/2019/57832019.
8. V. Sivapriya and R. Deebika, "Dimensionality Reduction Using F-Score Analysis Based on Support Vector," vol. 22, no. 1, 2016.
9. G. Orellana, B. Arias, M. Orellana, V. Saquicela, F. Baculima, and N. Piedra, "A study on the impact of pre-processing techniques in Spanish and english text classification over short and large text documents," *Proc. - 3rd Int. Conf. Inf. Syst. Comput. Sci. INCISCOS 2018*, vol. 2018-Decem, pp. 277–283, 2018, doi: 10.1109/INCISCOS.2018.00047.
10. R. A. Aziz *et al.*, "Two Stages Song Subject Classification on Indonesian Song Based on Lyrics, Genre & Artist," *2016 Int. Conf. Inf. Technol. InCITe 2016 - Next Gener. IT Summit Theme - Internet Things Connect your Worlds*, no. 05, pp. 21–24, 2018, doi: 10.1109/SIET.2018.8693201.
11. F. A. Ma'Ruf, Adiwijaya, and U. N. Wisesty, "Analysis of the influence of Minimum Redundancy Maximum Relevance as dimensionality reduction method on cancer classification based on microarray data using Support Vector Machine classifier," *J. Phys. Conf. Ser.*, vol. 1192, no. 1, 2019, doi: 10.1088/1742-6596/1192/1/012011.
12. M. Syamala and N.J.Nalini, "A Deep Analysis on Aspect based Sentiment Text Classification Approaches," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 4, no. 2, pp. 15–21, 2019, doi: https://doi.org/10.30534/ijatcse/2019/01852019.
13. A. K. Singh and M. Shashi, "Vectorization of Text Documents for Identifying Unifiable News Articles," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 7, pp. 305–310, 2019, doi: 10.14569/ijacsa.2019.0100742.
14. F. P. Shah and V. Patel, "A review on feature selection and feature extraction for text classification," *Proc. 2016 IEEE Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2016*, pp. 2264–2268, 2016, doi: 10.1109/WiSPNET.2016.7566545.
15. S. Srirangamsridharan, M. Srivatsa, R. Ganti, and C. Simpkin, "Doc2Img: A New Approach to Vectorization of Documents," *2018 21st Int. Conf. Inf. Fusion, FUSION 2018*, pp. 2172–2178, 2018, doi: 10.23919/ICIF.2018.8455685.
16. D. Isa, L. H. Lee, V. P. Kallimani, and R. Rajkumar, "Text document preprocessing with the bayes formula for classification using the support vector machine," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 9, pp. 1264–1272, 2008, doi: 10.1109/TKDE.2008.76.
17. L. H. Lee, C. H. Wan, R. Rajkumar, and D. Isa, "An enhanced Support Vector Machine classification framework by using Euclidean distance function for text document categorization," *Appl. Intell.*, vol. 37, no. 1, pp. 80–99, 2012, doi: 10.1007/s10489-011-0314-z.

18. O. J. Okesola, K. O. Okokpujie, A. A. Adewale, S. N. John, and O. Omoruyi, “An Improved Bank Credit Scoring Model: A Naïve Bayesian Approach,” *Proc. - 2017 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2017*, pp. 228–233, 2018, doi: 10.1109/CSCI.2017.36.
19. J. Rexiline Ragini and P. M. Rubesh Anand, “An empirical analysis and classification of crisis related tweets,” *2016 IEEE Int. Conf. Comput. Intell. Comput. Res. ICCIC 2016*, pp. 2–5, 2017, doi: 10.1109/ICCIC.2016.7919608.
20. D. Buzic and J. Dobsa, “Lyrics classification using Naive Bayes - IEEE Conference Publication,” *2018 41st Int. Conv. Inf. Commun. Technol. Electron. Microelectron.*, p. 5, 2018, doi: 10.3390/genes6020372.
21. M. Baygin, “Classification of Text Documents based on Naive Bayes using N-Gram Features,” *2018 Int. Conf. Artif. Intell. Data Process. IDAP 2018*, pp. 1–5, 2019, doi: 10.1109/IDAP.2018.8620853.
22. D. Leman, “Expert System Diagnose Tuberculosis Using Bayes Theorem Method and Shafer Dempster Method,” *2018 6th Int. Conf. Cyber IT Serv. Manag. CITSM 2018*, no. Citsm, pp. 1–4, 2019, doi: 10.1109/CITSM.2018.8674380.
23. N. Ramkumar, S. Prakash, S. A. Kumar, and K. Sangeetha, “Prediction of liver cancer using Conditional probability Bayes theorem,” *2017 Int. Conf. Comput. Commun. Informatics, ICCCI 2017*, pp. 7–11, 2017, doi: 10.1109/ICCCI.2017.8117752.
24. F. Ikorasaki and M. B. Akbar, “Detecting Corn Plant Disease with Expert System Using Bayes Theorem Method,” *2018 6th Int. Conf. Cyber IT Serv. Manag. CITSM 2018*, no. Citsm, pp. 1–3, 2019, doi: 10.1109/CITSM.2018.8674303.
25. C. Cassandra and R. Sari, “Agricultural Expert System Design Based on Bayes Theorem,” *Proc. 2018 Int. Conf. Inf. Manag. Technol. ICIMTech 2018*, no. September, pp. 315–320, 2018, doi: 10.1109/ICIMTech.2018.8528127.
26. X. Deng, J. Guo, Y. Chen, and X. Liu, “A method for detecting document orientation by using Naïve Bayes classifier,” *Proc. 2012 Int. Conf. Ind. Control Electron. Eng. ICICEE 2012*, pp. 429–432, 2012, doi: 10.1109/ICICEE.2012.120.
27. L. R. A. X. Menezes and A. J. F. Loureiro, “CmWave through vegetation: Correlation of pixels and attenuation using UT and Bayes Inference,” *2017 IEEE Antennas Propag. Soc. Int. Symp. Proc.*, vol. 2017-Janua, pp. 1829–1830, 2017, doi: 10.1109/APUSNCURSINRSM.2017.8072957.
28. P. Goodwin, “When simple alternatives to Bayes formula work well: Reducing the cognitive load when updating probability forecasts,” *J. Bus. Res.*, vol. 68, no. 8, pp. 1686–1691, 2015, doi: 10.1016/j.jbusres.2015.03.027.
29. H. Liu, Z. Liu, X. Wang, and Y. Cai, “Bayes’ Theorem based maritime safety information classifier,” *Proc. 30th Chinese Control Decis. Conf. CCDC 2018*, pp. 2725–2729, 2018, doi: 10.1109/CCDC.2018.8407588.
30. L. H. Lee, C. H. Wan, R. Rajkumar, and D. Isa, “An enhanced Support Vector Machine classification framework by using Euclidean distance function for text document categorization,” *Appl. Intell.*, vol. 37, no. 1, pp. 80–99, 2012, doi: 10.1007/s10489-011-0314-z.
31. L. H. Lee, R. Rajkumar, and D. Isa, “Automatic folder allocation system using Bayesian-support vector machines hybrid classification approach,” *Appl. Intell.*, vol. 36, no. 2, pp. 295–307, 2012, doi: 10.1007/s10489-010-0261-0.
32. S. Aggarwal and D. Kaur, “Enhanced Smoothing Methods Using Naïve Bayes Classifier for Better Spam Classification,” vol. 2, no. 9, pp. 3061–3073, 2013.
33. S. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” *Comput. Speech Lang.*, vol. 13, no. 4, pp. 359–393, 1999.
34. K. P. Murphy, “Naive Bayes Classifier,” 2006.
35. M. H. G, “Metode Smoothing dalam Naive Bayes untuk Klasifikasi Email Spam,” in *Bogor: Institut Pertanian Bogor.*, 2014.
36. Q. Yuan, G. Cong, and N. M. Thalmann, “Enhancing Naive Bayes with various smoothing methods for short text classification,” in *WWW’12 - Proceedings of the 21st Annual Conference on World Wide Web Companion*, 2012, pp. 645–646, doi: 10.1145/2187980.2188169.
37. Y. G. Jung, K. T. Kim, B. Lee, and H. Y. Youn, “Enhanced Naive Bayes Classifier for real-Time sentiment analysis with SparkR,” *2016 Int. Conf. Inf. Commun. Technol. Converg. ICTC 2016*, pp. 141–146, 2016, doi: 10.1109/ICTC.2016.7763455.
38. P. V Arivoli and T. Chakravarthy, “Document Classification Using Machine Learning Algorithms - A Review,” vol. 5, no. 2, pp. 48–54, 2015, [Online]. Available: www.ijser.in.
39. A. Wibowo Haryanto, E. Kholid Mawardi, and Muljono, “Influence of Word Normalization and Chi-Squared Feature Selection on Support Vector Machine (SVM) Text Classification,” *Proc. - 2018 Int. Semin. Appl. Technol. Inf. Commun. Creat. Technol. Hum. Life, iSemantic 2018*, pp. 229–233, 2018, doi: 10.1109/ISEMANTIC.2018.8549748.
40. R. Y. Goh and L. S. Lee, “Credit Scoring: A Review on Support Vector Machines and Metaheuristic Approaches,” *Adv. Oper. Res.*, vol. 2019, 2019, doi: 10.1155/2019/1974794.
41. H. T. Sueno, B. D. Gerardo, and R. P. Medina, “Transforming Text Documents Into Numerical Format Using Enhanced Bayesian Vectorization For Multi-class Classification,” no. 3, pp. 18–22, 2020.
42. A. Cardoso-Cachopo, “Datasets for single label text categorization,” *Artificial Intelligence Group, Department of Information Systems and Computer Science, Instituto Superior Tecnico, Portugal.*, 2011. http://web.ist.utl.pt/~acardoso/datasets/.
43. R. Misra, “News Headlines Dataset For Sarcasm

Detection," doi: 10.13140/RG.2.2.16182.40004.

44. N. Sharma and M. Singh, "Modifying Naive Bayes classifier for multinomial text classification," *2016 Int. Conf. Recent Adv. Innov. Eng. ICRAIE 2016*, 2017, doi: 10.1109/ICRAIE.2016.7939519.
45. A. Javed, "Unfolding Naïve Bayes from Scratch," *Towards Data Science*, 2018. https://towardsdatascience.com/unfolding-naïve-bayes-from-scratch-2e86dcae4b01 (accessed Aug. 12, 2019).
46. C. C. Chang and C. J. Lin, "LIBSVM: A Library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–39, 2011, doi: 10.1145/1961189.1961199.