Analysis of Word Vector Representation Techniques with Machine-Learning Classifiers for Sentiment Analysis of Public Facebook Page's Comments in Myanmar Text

Hay Mar Su Aung
University of Computer Studies
Yangon, Myanmar
haymarsuaung@ucsy.edu.mm

Win Pa Pa
University of Computer Studies
Yangon, Myanmar
winpapa@ucsy.edu.mm

Abstract

This paper presents a study of comparison on three different machine learning techniques to sentiment analysis for Myanmar language. The fundamental part of sentiment analysis (SA) is to extract and identify the subjective information that is social sentiment in the source text. The sentiment class is positive, neutral or negative of a comment. The experiments are done on the collected 10,000 Facebook comments in Myanmar language. The objective of this study is to increase the accuracy of the sentiment identification by using the concept of word embeddings. Word2Vec is used to train for producing high-dimensional word vectors that learns the syntactic and semantic of word. The resulting word vectors train Machine Learning algorithms in the form of classifiers for sentiment identification. This experimental results prove that the use of word embeddings from the collected real world datasets improved the accuracy of sentiments classification and Logistic Regression outperformed the other two ML methods in terms of accuracy and F-measures.

Keywords— *Multiclass classification, natural language processing, sentiment analysis, Facebook Page's comments, word embedding, Logistic Regression.*

I. INTRODUCTION

Sentiment Analysis (SA) is a type of contextual mining of text that identifies and extracts the tendency of people's feelings via Natural Language Processing, computational linguistics and text analysis, which are helped to use extracting and analyzing subjective information from the website majority social media and similar sources. It is used to quantify the general public's social attitude of specific brand, product or service when it monitors online conversations. Sentiment Analysis (SA) is also known as opinion mining. SA uses data mining techniques to extract and capture analyzed data to classify the opinion of a document or collection of documents, like blog posts, reviews of product and social media feeds like Tweets, status updates and comments.

In the distinctive events of technology, people stored data from the internet. These storing data have been grown every day and the large amount of data is stored up to date. But very vital information is carried by these data carry concerning the emotions of different people around the world, so it has become essential to summarize these large number of data with particular automated systems. In this day, many people use social media around the world like Twitter, Facebook and so on. Among of them, Facebook is a popular free social networking website. In Myanmar, most people use Facebook to express their opinions and feelings.

This paper analysis the performance of the word vector techniques (Word2Vec , TFIDF text feature vector representation and pre-trained Word2Vec) with three different machine learning techniques (Logistic Regression, SVM and Random Forest) for public Facebook Page's comments of Myanmar text.

This paper is constructed as the following. The related techniques of word vector is exhibited as Section 2. Section 3 describes the details of proposed system. The experiment results is explained at the Section 4. Finally, Section 5 concludes the discussion with possible enhancements to the proposed system.

II. RELATED TECHNIQUES

A. Word Embeddings

A word embedding learns the text representation where words that have the same meaning have a similar representation. It is emerged

Word Embedding in the Natural Language Processing (NLP) field. Word Embeddings, a text mining technique, is to establish association between words in the set of sentences. The syntactic and semantic meanings of words are comprehend from the context. The idea of distributional assumption propose that words occurring in the alike words are semantically alike. There are the two techniques of word embeddings – (a) Frequency Based Embeddings (b) Prediction Based Embeddings. The Frequency Based Embeddings processed poorly at conserving the contextual information in textual data such as the traditional bag-of-words model. The Prediction Based Embeddings predicts a target word from the given a context word. The researchers developed Global Vectors for Words Representation (GloVe) which is the algorithm of an unsupervised learning to achieve word vector representation does very well at context preservation.

B. Word2Vec

Word2Vec is the producing of the model for word embeddings from a set of sentences for word representation.Word2Vec represents content of vectors into vector space. Word2Vec models use shallow two-layer neural networks, consists of one hidden layer (projection layer) between input and output, which are used to train for reconstructing linguistic contexts of words. Word2Vec takes a set of sentences (corpus) as its input and produces word vectors in a vector space, generally several hundred of dimensions, with each unique word in the text corpus being placed a corresponding vector in the space. Word vectors are placed in the vector space in such a way the words that have similar contexts in the set of sentences are appeared in approximately close to each other in the space. In the same way, different meaning of word contents are located far away from each other. Word2Vec have two techniques to achieve word vectors. They are (i) Continuous Bag of words (C-BOW model) and (ii) Skip-Gram model as shown in Figure 1 and Figure 2. The C-BOW model is used to train for predicting the current word based on its context words. The Skip-Gram model is used to train the model for predicting the context words given a current word. w(t) is the context word and [w(t-2),w(t-1),w(t+1) and w(t+2)] are the surrounding words.

C. tfidf Vectorization

tfidf is an shorthand for term frequency-inverse document frequency. The algorithm of tfidf is a very common algorithm for converting word into a meaningful numeric representation. tfidf vector is based on the concurrency apporach. It is different from the count vectorization approach because it takes into account not just the number of times a text in a single document but in all documents of the corpus. The technique is widely used as features extraction across various applications in NLP field. tfidf need to assign weight for the word based on the number of a word that occurs in the document also consideration the occurrences of the word in all documents.

tf determines the frequency of word appears in a specific document. The calculation equation is as:

$$tf = \frac{\text{occurencies of a word in the document}}{\text{total words in the document}} \quad (1)$$

Inverse Document Frequency for a specific word measures the log of the division of the sum of all document numbers and the document numbers with concluding the particular word in it. It is calculated as the following.

$$idf = log \frac{\text{Total number of docs}}{\text{Number of docs} \in \text{the word}} \quad (2)$$

$$tf - idf = tf * idf \quad (3)$$

tfidf is needed to transform from the specific words to numerical feature vectors.

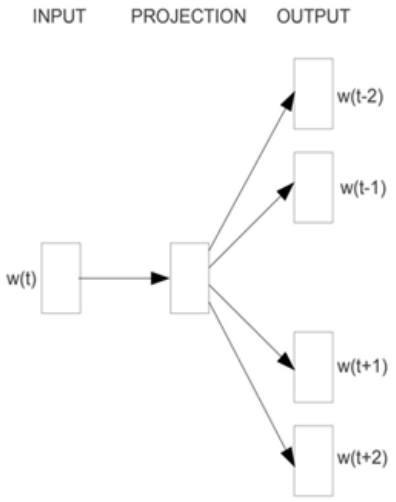
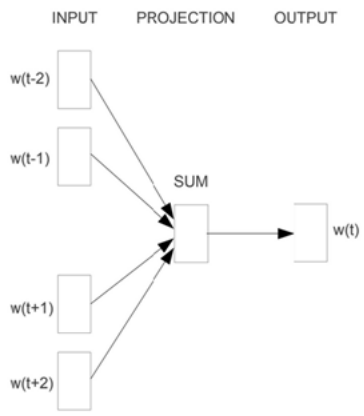


Figure 1. Architecture of C-BOW Model



I.

Figure 2. Architecture of Skip-Gram Model

III. METHODOLOGY

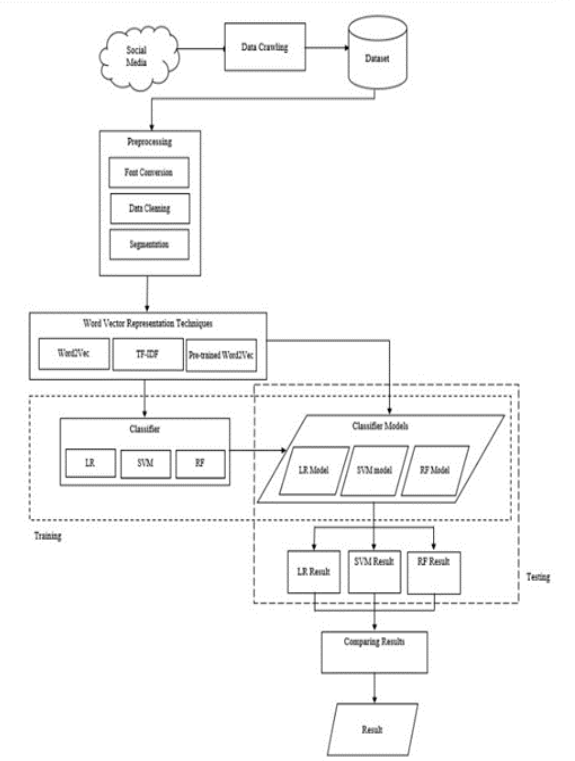


Figure 3. Proposed System Design

The following steps are processed in this proposed system.

A. Data Collecting

There is no previously created dataset for public Facebook page comments in Myanmar language. So, the dataset is crawled public comments of Facebook page related to "Myanmar Celebrity" Page in Myanmar to create own dataset. Social media data (Facebook comments) is collected through data crawling using Facepager tool. These comments were copied and saved as two text file (training text file and testing text file). This corpus is comprised of posts over 100 and comments of 10,000. The set of sentences are antecedently identified with the labels for indicating the user emotions each Facebook comment. The following figure show the visual analysis of imbalance between the three sentiment classes of the dataset. There are 10,000 sentences training dataset in which 2,316 sentences labeled by negative, 1,461 sentences labeled by neutral and 6,223 sentences labeled by positive. There are 1,000 sentences testing data in which 345 sentences labeled by negative, 148 sentences labeled by neutral and 507 sentences labeled by positive.

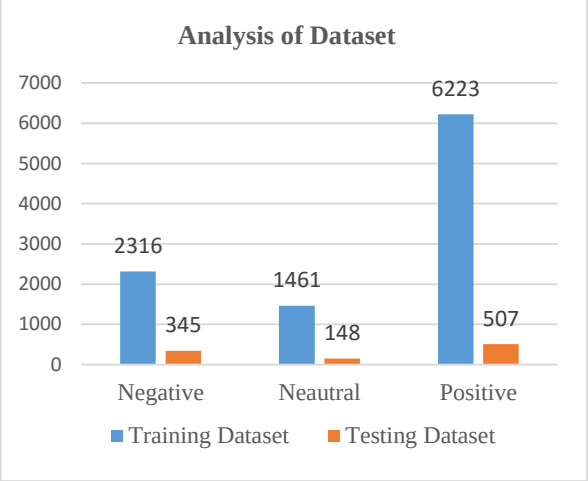


Figure 4. Analysis of Facebook comments Dataset

B. Data Preprocessing

- Font Conversion

  Most of Facebook user in Myanmar use Zawgyi font on social media. Application in technology field use Unicode. In this paper, the collected text data are both Zawgyi and Unicode format. So, the gather text data with Zawgyi format are converted to Unicode by using online Zawgyi-Unicode converter.

- Data Cleaning

  The collected text comments in the dataset consist of various kinds of noises. Before the training and testing are proceeded, these noises are removed all undesirable marks of punctuation, hash sign, additional spaces, unrecognised characters, number etc. from the set of sentences. Most of Facebook users tag the name in the comments so this tagging name is also removed. Due to occur noise in the corpus, the comments which may take place more than one time in the corpus are taken as one.

- Segmentation

  Word segmentation is a necessary part before to process Natural Language Processing in the Myanmar text, because a text of Myanmar language is a string of characters from left to right without explicit word boundary elimiters. It is firstly needed to segment text into separate meaningful words in order to extract words from the collected dataset in Myanmar text. In this paper, Myanmar text data are manually segmented.

TABLE I. EXAMPLE OF SEGMENTATION THE COMMENTS

အမြဲအားပေးနေပါတယ်ချစ်တဲ့မ (I'm always encouraging lovely Ma) အမြဲ အားပေး နေ ပါ တယ် ချစ် တဲ့ မ	Positive
ဘာရေးရမှန်းမသိတော့ဘူး😂😂😂 (I don't know what to write 😂😂😂) ဘာ ရေး ရ မှန်း မ သိ တော့ ဘူး 😂 😂 😂	Neutral
Mcကိုစိတ်ပျက်လာပီ (Mc is disappointed) Mc ကို စိတ်ပျက် လာ ပီ	Negative

C. Word Vector Representation Method

- Building Word2Vec Model

  The study of this paper use C-BOW model of the Word2Vec algorithm. The implementation of Word2Vec in the Gensim python library was used to train on the treated data. For achieving the better implementation of Word Embeddings, most consideration is needed to give certain hyper-parameters. These parameters are: training dataset, dimensionality, context-window, minimum word count, sub-sampling and iteration. The training dataset trained 10,000 sentences of comments (training data). Negative sampling was set because it proved to be efficiently computational relative to hierarchical softmax. For resulting in the better Word Embeddings, the projection layer of the neural network was assigned 300. The context-window size was used 50 to prescribe context-window size for C-BOW model. 1e-3 was used the sub-sampling rate to counter the imbalance dataset of frequent words. 1 set as the minimum count for considering each word in the set of sentences during processing of training. It trained Word2Vec model on the treated data with the above mentioned settings of hyper-parameter and stored in a data file format. The word embedding model produced d*p dimension word vectors where d is the dictionary size and p is the projection layer (hidden layer) size. In this study, the size of the dictionary for the training set is 5,833 vocabulary in the dictionary.

- Building tfidf Vectorization

  tfidf implementation of sklearn python library was trained on the data for training. The same training dataset was used to take as its input corpus. The word embedding model produce word vectors having r×u dimension where r is the number of row in the given training data and u is the unique words from all text. In this study, the rare words is 319 words.

- Building Pre-trained Word2Vec Model

  For pre-trained Word2Vec, the pre-trained C-BOW model of Word2Vec are previously created by using another dataset containing 750,000 sentences. The training data and testing data may or may not contain in this dataset. The pre-trained Word2Vec model is similarly created with the Word2Vec algorithm. In this paper, the 300 dimensions of this model is assigned. The context window of 5 was used for C-BOW models with workers of 10 (the number of processor). And then, the generated pre-trained word vector model is the vocabulary size of 49,740 unique words with 300 dimensions.

D. Classification

In the field of the sentiment analysis, supervised learning is useful for training the data on a pattern that may analyze for either the opinion is positive, negative or neutral. In this paper, we have chosen the three Machine Learning classification techniques consisting of Logistic Regression (LR), Support Vector Machine (SVM) and Random Forest (RF).

IV. EXPERIMENTAL RESULTS

The analysis has been performed comparative according to examine the performance of the three

techniques of classification: Logistic Regression, SVM and Random Forest. In the sentiment classification field, the evaluation of the techniques have used the common information retrieval metrics; precision, recall and f-score. According to evaluate the process of sentence classification, an information retrieval task is considered. The precision can be calculated as (4):

$$Precision = \frac{TP}{(TP+FP)} \quad (4)$$

In which, *TP* is true positive (the occurrences of targets actually identified sentences) and *FP* is false positive (the occurrences of targets that were not correctly identified sentences). Moreover, recall can be calculated as (5):

$$Recall = \frac{TP}{(TP+FN)} \quad (5)$$

In which, *TP* is true positive (the occurrences of targets actually identified sentences) and *FN* is false negative (the occurrences of sentences which were not identified at all). It is possible to calculate the f-score as follows (6):

$$F1 - score = \frac{2*(Precision*Recall)}{(Precision+Recall)} \quad (6)$$

A. Logistic Regression

Logistic Regression classifier is used to apply with the proposed features vectors extraction (tfidf vectorizer, Word2Vec and pre-trained Word2Vec). Using the confusion metrics which are precision (Positive Predictive Value), recall (True Positive Rate) and F-score, the evaluation has been performed. The following tables show the results.

In TABLE II, the evaluation of classification details how the logistic regression classifier with tfidf word vector representation technique carried out for predicting every sentiment label. In the negative label, the classifier exactly prophesied TPR of 38% with the PPV of 63% and F1-score of 47% in supporting 345 negative test objects. In the neutral label, the classifier exactly prophesied TPR of 0.1% with the PPV 44% and F1-score of 16% in supporting 148 neutral test objects. In the positive label, the classifier exactly predict 91% of 507 positive test objects, properly with the PPV of 61% and F1-score of 73%. Logistic Regression classifier with tfidf vectorizer also tried to precisely predict like the other training algorithms. In TABLE III, many objects that were not correctly prophesied label as positive. For instance, according to the Table 3, 203 out of 345 negative comments were predicted to be positive while 92 out of 148 neutral comments were predicted to be positive.

TABLE II. EVALUATION OF LOGISTIC REGRESSION WITH TFIDF VECTORIZER

	PPV	TPR	F1-score	Support
Negative	0.63	0.38	0.47	345
Neutral	0.44	0.1	0.16	148
Positive	0.61	0.92	0.73	507
Average/ Total	0.59	0.61	0.56	1000

TABLE III. CONFUSION MATRIX FOR LOGISTIC REGRESSION WITH TFIDF VECTORIZER

Negative	130	12	203
Neutral	42	14	92
Positive	36	6	465
	Negative	Neutral	positive

In TABLE IV, the evaluation of classification details the Word2Vec+Logistic Regression classifier with carried out for predicting every sentiment class. In the negative label, the classifier exactly prophesied 82% of rate 77% and F1-score of 80%. In the neutral label, the classifier exactly prophesied 38% of 148 neutral test objects, properly with the PPV of 63% and F1-score of 47%. In the positive label, the classifier exactly prophesied 93% of 507 positive test objects, properly with the PPV of 86% and F1-score of 89%. This approach also tried to fit for precisely classifying like other training algorithms. In TABLE V, 44 out of 345 negative comments were predicted to be positive while 57 out of 148 neutral comments were predicted to be negative.

TABLE IV. EVALUATION OF LOGISTIC REGRESSION WITH WORD2VEC

	PPV	TPR	F1-score	Support
Negative	0.788	0.82	0.804	345
Neutral	0.629	0.378	0.473	148
Positive	0.857	0.933	0.893	507
Average/ Total	0.80	0.81	0.80	1000

TABLE V. CONFUSION MATRIX FOR LOGISTIC REGRESSION WITH WORD2VEC

Negative	283	18	44
Neutral	57	56	35
Positive	19	15	473
	Negative	Neutral	Positive

In TABLE VI, the evaluation of classification details how Pre-trained Word2Vec word vector representation technique + the Logistic Regression classifier carried out to predict every sentiment class. In the negative label, the classifier exactly prophesied 77% of 345 negative test objects properly with the PPV of 77% and F1-score of 77%. For the neutral class, the classifier exactly prophesied 38% of 148 neutral test objects properly with the PPV of 53% and F1-score of 44%. For the positive class, the classifier exactly prophesied 92% of 507 positive test objects properly with the PPV of 85% and F1-score of 88%.

TABELE VI. EVALUATION OF LOGISTIC REGRESSION WITH PRE-TRAINED WORD2VEC

	PPV	TPR	F1-score	Support
Negative	0.765	0.774	0.769	345
Neutral	0.533	0.378	0.443	148
Positive	0.85	0.915	0.881	507
Average/ Total	0.77	0.79	0.78	1000

TABLE VII. CONFUSION MATRIX FOR LOGISTIC REGRESSION WITH PRE-TRAINED WORD2VEC

Negative	267	33	45
Neutral	55	56	37
Positive	27	16	464
	Negative	Neutral	Positive

B. Support Vector Machine (SVM)

SVM classifier is applied with the tfidf vectorizer, Word2Vec and pre-trained Word2Vec. Using the confusion metrics which are precision (Positive Predictive Value), recall (True Positive Rate) and F-score, the evaluation has been performed. The following tables show the results.

In TABLE VIII, the evaluation of classification details how the SVM classifier with tfidf word vector representation technique carried out for each sentiment class. In the negative label, the classifier exactly prophesied 37% of 345 negative test objects properly with 60% of the PPV and F1-score of 46%. In the neutral label, the classifier definitely prophesied 0.4% of 148 neutral test objects correctly with 35% of the PPVand F1-score of 0.7%. In the positive label, the classifier exactly prophesied 91% of 507 positive test objects correctly with the PPVof 60% and F1-score of 72%. Logistic Regression classifier with tfidf vectorizer also struggled to precisely classify like other training algorithms. As shown in TABLE IX, many objects that were not correctly prophesied were labeled as positive. For instance, in TABLE IX, 212 out of 345 negative comments were predicted to be positive while 99 out of 148 neutral comments were predicted to be positive.

TABLE VIII. EVALUATION OF SVM WITH TFIDF VECTORIZER

	PPV	TPR	F1-score	Support
Negative	0.604	0.371	0.46	345
Neutral	0.353	0.041	0.073	148
Positive	0.597	0.907	0.72	507
Average/ Total	0.56	0.59	0.53	1000

TABLE IX. CONFUSION MATRIX SVM WITH IFIDF VECTORIZER

Negative	128	5	212
Neutral	43	6	99
Positive	41	6	460
	Negative	Neutral	Positive

In TABLE X, the evaluation of classification details how the SVM classifier with Word2Vec word vector representation technique processed for predicting every sentiment label. In the negative label, the classifier definitely prophesied 82% of 345 negative test objects properly with the PPV of 78% and F1-score of 80%. In the neutral label, the classifier definitely prophesied 34% of 148 neutral test objects properly with the PPV of 56% and F1-score of 42%. In the positive label, the classifier definitely prophesied 92% of 507 positive test objects correctly

with the PPV of 85% and F1-score of 89%. SVM classifier with Word2Vec also struggled to precisely classify like other training algorithms. According to the TABLE XI, 42 out of 345 negative comments were prophesied to be positive while 60 out of 148 neutral comments were prophesied to be negative.

TABLE X. EVALUATION OF SVM WITH WORD2VEC

	PPV	TPR	F1-score	Support
Negative	0.775	0.817	0.795	345
Neutral	0.556	0.338	0.42	148
Positive	0.853	0.919	0.885	507
Average/ Total	0.78	0.80	0.79	1000

TABLE XI. CONFUSION MATRIX FOR SVM WITH WORD2VEC

Negative	282	21	42
Neutral	60	50	38
Positive	22	19	466
	Negative	Neutral	Positive

In TABLE XII, the evaluation of classification details how the SVM classifier with Pre-trained Word2Vec word vector representation technique carried out predicting for every sentiment label. In the negative label, the classifier definitely prophesied 80% of 345 negative test objects correctly with the PPV of 75% and F1-score of 77%. In the neutral label, the classifier definitely prophesied 32% of 148 neutral test objects properly with the PPV of 53% and F1-score of 40%. In the positive label, the classifier definitely prophesied 92% of 507 positive test objects properly with the 85% of PPV and F1-score of 88%.

TABLE XII. EVALUATION OF SVM WITH PRE-TRAINED WORD2VEC

	PPV	TPR	F1-score	Support
Negative	0.751	0.797	0.774	345
Neutral	0.533	0.324	0.403	148
Positive	0.853	0.915	0.883	507
Average/ Total	0.77	0.79	0.77	1000

TABLE XIII. CONFUSION MATRIX FOR SVM WITH PRE-TRAINED WORD2VEC

Negative	275	26	44
Neutral	64	48	36
Positive	27	16	464
	Negative	Neutral	Positive

C. Random Forest

Random Forest classifier is applied with the proposed features extraction (tfidf vectorizer, Word2Vec and pre-trained Word2Vec). Using the confusion metrics which are precision (Positive Predictive Value), recall (True Positive Rate) and F-score, the evaluation has been performed. The following tables show the results.

In TABLE XIV, the evaluation of classification details how the Random Forest classifier with tfidf word vector representation technique carried out for every sentiment label. In the negative label, the classifier definitely prophesied 36% of 345 negative test objects properly with the PPV of 60% and F1-score of 45%. In the neutral label, the classifier definitely prophesied 10% of 148 neutral test objects properly with the PPV of 26% and F1-score of 15 %. In the positive label, the classifier definitely prophesied 90% of 507 positive test objects properly with the 62% of PPV and F1-score of 73%. Logistic Regression classifier with tfidf vectorizer also struggled to precisely classify like other training algorithms. As TABLE XV, many objects that were not correctly prophesied were labeled as positive. For instance, in TABLE XV, 203 out of 345 negative comments were prophesied to be positive while 92 out of 148 neutral comments were prophesied to be positive.

TABLE XIV. EVALUATION OF RANDOM FOREST WITH TFIDF VECTORIZER

	PPV	TPR	F1-score	Support
Negative	0.6	0.357	0.447	345
Neutral	0.263	0.101	0.146	148
Positive	0.617	0.9	0.731	507
Average/ Total	0.56	0.59	0.55	1000

TABLE XV. CONFUSION MATRIX FOR RANDOM FOREST WITH TFIDF VECTORIZER

Negative	130	12	203
Neutral	42	14	92
Positive	36	6	465
	Negative	Neutral	Positive

In TABLE XVI, the evaluation of classification details how the Random Forest classifier with Word2Vec word vector representation technique carried out for every sentiment label. In the negative label, the classifier definitely prophesied 78% of 345 negative test objects correctly with the PPV of 72% and F1-score of 75%. In the neutral label, the classifier definitely prophesied 12% of 148 neutral test objects properly with the PPV of 43% and F1-score of 20%. In the positive label, the classifier definitely prophesied 93% of 507 positive test objects properly with the PPV of 81% and F1-score of 93%. Random Forest classifier with Word2Vec also struggled to precisely classify like other training algorithms. According to TABLE XVII, 58 out of 345 negative comments were prophesied to be positive while 76 out of 148 neutral comments were prophesied to be negative.

TABLE XVI. EVALUATION OF RANDOM FOREST WITH WORD2VEC

	PPV	TPR	F1-score	Support
Negative	0.717	0.78	0.747	345
Neutral	0.429	0.122	0.189	148
Positive	0.808	0.929	0.864	507
Average/ Total	0.72	0.76	0.72	1000

TABLE XVII. CONFUSION MATRIX FOR RANDOM FOREST WITH WORD2VEC

Negative	269	18	58
Neutral	76	18	54
Positive	30	6	471
	Negative	Neutral	Positive

In TABLE XVIII, the evaluation of classification details how the Random Forest classifier with Pre-trained Word2Vec word vector representation technique carried out for every sentiment label. In the negative label, the classifier definitely predicted 75% of 345 negative test objects properly with the PPV of 71% and F1-score of 73%. In the neutral label, the classifier definitely prophesied 17% of 148 neutral test objects correctly with the PPV of 46% and F1-score of 24%. In the positive label, the classifier definitely prophesied 92% of 507 positive test objects correctly with the PPV of 80% and F1-score of 86%.

TABLE XVIII. EVALUATION OF RANDOM FOREST WITH PRE-TRAINED WORD2VEC

	PPV	TPR	F1-score	Support
Negative	0.707	0.748	0.727	345
Neutral	0.463	0.169	0.2448	148
Positive	0.8	0.917	0.855	507
Average/ Total	0.72	0.75	0.72	1000

TABLE XIX. CONFUSION MATRIX FOR RANDOM FOREST WITH PRE-TRAINED WORD2VEC

Negative	258	20	67
Neutral	74	25	49
Positive	33	9	465
	Negative	Neutral	Positive

V. CONCLUSION

The paper of this study compares the performance of 3 different Machine Learning (ML) techniques containing Logistic Regression, SVM and Random Forest while using Myanmar sentiment analysis based on word vector representation method. The dataset that has been used is a collection of Myanmar Facebook pages comments for the purposes of sentiment analysis. The experimental results have proved that Logistic Regression classifier with Word2Vec has exceeded in performance than the other two Machine Learning by obtaining **80%** of F1-score. Hence, we can draw to conclude that given a word vector representation from Word2Vec instead of tfidf and pre-trained Word2Vec for extracting feature vectors, sentiment analysis in Myanmar on three labels of ideas, will carried out better the use of Logistic Regression than the use of SVM and Random Forest.

REFERENCES

[1] Aye Myat Mon, Khin Mar Soe. “Clustering Analogous Words in Myanmar Language using Word Embedding Model”, ICCA & ICFCC 2019 Conference, 27th February 2019.

[2] C. Emelda. “A comparative Study on Sentiment Classification and Ranking on Product Reviews”, ijirae, issn: 2349-2163, volume 1 issue 10, November 2014.

[3] Merfat M. Altawaier, Sabrina Tiun, “Comparison of Machine Learning Approaches on Arabic Twitter Sentiment Analysis”, volume.6 (2016) no. 6, ISSN: 2088-5334

[4] Md. Al- Amin, Md. Saiful Islam, Shapan Das Uzzal, "Sentiment Analysis of Bengali Comments With Word2Vec and Sentiment Information of Words", 978-1-5090-5627-9/17/$31.00 ©2017 IEEE.

[5] Joshua Acosta, Norissa Lamaute, Mingxiao Luo, Ezra Finkelstein, and Andreea Cotoranu. “Sentiment Analysis of Twitter Messages Using Word2Vec”. Proceedings of Student-Faculty Research Day, CSIS, Pace University, May 5th, 2017.

[6] Oscar B. Deho, William A. Agangiba,Felix L. Aryeh, Jeffery A. Ansah. “Sentiment Analysis with Word Embedding”.

[7] Soe Yu Maw, May Aye Khine, “Aspect based Sentiment Analysis for travel and tourism in Myanmar Language using LSTM”, ICCA & ICFCC 2019 Conference Program Schedule 27th February 2019.

[8] https://towardsdatascience.com/updated-text-preprocessing-techniques-for-sentiment-analysis-549af7fe412a

[9] https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/

[10] https://machinelearningmastery.com/what-are-word-embeddings/

[11] https://hackernoon.com/word-embeddings -in-nlp-and-its-applications-fab15eaf7430

[12] https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/

[13] https://medium.com/data-science-group-iitr/word-embedding-2d05d270b285

[14] https://medium.com/@shiivangii/data-representation -in-nlp-7bb6a771599a