Systematic Comparison of Vectorization Methods in Topic Classification for Myanmar Language

Tun Ye Minn
University of Information Technology, Yangon,
tunyeminn@uit.edu.mm

Aung Khant Myat
University of Information Technology, Yangon,
aungkhantmyat@uit.edu.mm

Khin Nyo Nyo Theint
University of Information Technology, Yangon,
khinnyonyotheint@uit.edu.mm

Thinn Thinn Wai
Faculty of computer Science
University of Information Technology, Yangon,
thinnthinnwai@uit.edu.mm

Hmway Hmway Tar
Faculty of Computer Science
University of Information Technology,
Yangon,
hmwaytar@uit.edu.mm

Win Win Thant
Faculty of Computer Science
University of Information Technology,
Yangon,
winwinthant@uit.edu.mm

Thet Thet Zin
Faculty of Computer Science
University of Information Technology, Yangon,
thetthetzin@uit.edu.mm

***Abstract*— Text classification which is a research area of text data mining is widely used and one of the popular tasks of Natural Language Processing (NLP). But very few had focused on the effective use of vectorization methods which are vital in classification systems. In this paper, three widely used vectorizers for topic classification are discussed. The main goal is to evaluate the effectiveness of three different vectorization techniques such as TF-IDF Vectorizer (TFID), Count Vectorizer, and Word2Vec Vectorizer in the context of Myanmar news topic classification. The study focuses on a dataset comprising Myanmar news articles categorized into four distinct topics: Politics, Entertainment, Crime, and Business. Each category has 180 unique articles and in total 720 articles are prepared. Furthermore, these articles undergo segmentation and preprocessing. Each vectorization technique is applied through experimentation on an RBF (Radial Basis Function) Support Vector Machine (SVM) model. Moreover, feature selection via Chi Square algorithm is applied to optimize performance. Results indicate that TFID outperforms the other vectorizers, as seen by its accuracy of 83% on the test set, compared to 75% for the Count vectorizer and 48% for the Word2Vec. Overall, this research contributes insights into the effective utilization of NLP techniques for Myanmar news classification by emphasizing the advantage of TFID Vectorization compared to other vectorization techniques.**

Keywords*— *Text Classification, Natural Language Processing, TFID vectorizer, Count vectorizer, Word2Vec vectorizer, Support Vector Machine, RBF kernel

I. INTRODUCTION

Effective NLP methods are becoming more and more important nowadays, particularly in the news classification domain. Although several machine learning (ML) techniques have been researched in the past for Myanmar news categorization tasks, few of them have really addressed the effective use of vectorization techniques.

Thus, this paper attempts to demonstrate the effectiveness of news classification through the use of unique vectorization techniques. Vectorization is the process of modifying text data into numerical representations which lies at the core of many NLP tasks. This study compares three vectorization methods: Word2Vec, Count Vectorizer, and TFID Vectorizer. The primary objective is to determine the most effective vectorizer for the Myanmar news topic classification. As part of the research process, these vectorization techniques are carefully contrasted and compared using an RBF SVM model. The model is chosen because it performs well in non-linear classification tasks and robustness for high-dimensional feature spaces.

In NLP tasks, especially in classification, the choice of vectorization is vital as it has a huge influence on the nature of the task. In fact, the choice of the vectorization technique is as important as the choice of the model for classification tasks. This paper showcases the different vectorization techniques and their impact on the accuracy of the news classification system.

News data is mainly collected from news media websites. The collected data is first labeled and then used as a training data set. To select the best data features, Chi-Square feature selection is applied to the proposed classification algorithm for the task. Finally, each vectorization technique is evaluated using performance metrics like accuracy, F1-score, and their respective recall and precision on SVM.

The remaining parts of the paper are structured in the following order. In section 2, the related works are explained and the Myanmar language's nature can be found in section 3. In section 4, not only the paper's proposed system overview is shown, but also the different types of Vectorizers and RBF SVM ML algorithm for text classification are explained. Experimental work is described in section 5 and finally, in sections 6 and 7, the conclusion and references are stated.

II. RELATED WORK

Numerous vectorizations have been employed for Automated Text Classification. The majority of the research has focused on English, and other Latin languages and models but very few studies have been carried out on

Myanmar text and different types of vectorization techniques.

For the Myanmar Language, in the paper [1], T.T.Zaw and K.M.Soe proposed the Myanmar language text classification using the TFID Genetic Algorithm Classifier. To select the related features, the term frequency-inverse document frequency (tf_idf) algorithm was used. With this algorithm, the proposed system classifier achieved a 90% accuracy rate according to the experimental results. K.T.Nwet, K.M.Soe, and A.H.Khine, from the University of Computer Studies, Yangon have developed an Automatic Myanmar News Classifier using supervised Methods. The feature selection method is applied using Chi Square and applied Classification Algorithms were Naïve Bayes and k-Nearest Neighbors (KNN). They developed a Myanmar News corpus which consists of 4 categories with total of 1200 documents. Results showed that the K-NN classifier is better than the naïve Bayesian classifier. However, they concluded that the K-NN has a problem with time complexity because its complexity is high. But, K-NN has better accuracy than other algorithms [2].

H. Aung and W. P. Pa, from the University of Computer Studies, Yangon have proposed a paper [3] that compares three different machine learning techniques for sentiment analysis in the Myanmar language. They have collected 10,000 Facebook comments in the Myanmar language. They aimed to demonstrate the performance of word vector techniques such as Word2Vec, TFIDF text feature vector representation, and pre-trained Word2Vec with three different machine learning techniques such as Logistic Regression, SVM, and Random Forest. Their experimental result showcased that the Logistic Regression classifier with the Word2Vec vectorization technique has better performance than the other two Machine Learning techniques and word vector techniques. They obtained 80% of the F1-score in this research. However, the paper compared two vectorization techniques mainly. In this paper, three different types of vectorizations and their pros and cons will be demonstrated for Myanmar News Topic Classification with the SVM model.

The study by U. Gumińska, A. Poniszewska-Maranda and J. Ochelska-Mierzejewska systematically compared different text vectorization methods with the focus of highlighting their influence on classification performance. The paper [4] utilized vectorization methods such as Continuous Bag of Words (CBOW), Skip-gram, TF-IDF, and Distributed Memory of Paragraph Vectors (DM-PV). Each of the methods is applied to two classification algorithms namely k-Nearest Neighbor (k-NN) and Naive Bayesian Classifier (NBC). The experiment results proved that CBOW with the TF-IDF feature extraction method showed the highest accuracy on both k-NN and NBC. DM-PV required fewer training epochs to demonstrate significant efficiency gains. The paper suggested that whole-text vectorization methods, like DM-PV, required fewer training epochs and demonstrated higher computational efficiency compared to word-based methods. The findings underscore the importance of selecting appropriate vectorization techniques for optimizing both classification accuracy and computational efficiency. However, this approach has not been done in the Myanmar Language for classification tasks.

In this paper, different vectorization techniques such as TFID, Word2Vec, and CountVec's performance will be compared with the SVM machine learning algorithm by utilizing the pre-classified training document sets. The aim is to make a comparative study between the vectorization techniques on a Myanmar data set and find the optimal vectorization method for topic classification NLP task.

III. OVERVIEW OF THE PROPOSED SYSTEM

In this system, 4 data categories namely crime, business, entertainment, and politics are defined with 3 different vectorization techniques. Since the supervised learning method is utilized, raw data is required to be collected for training.

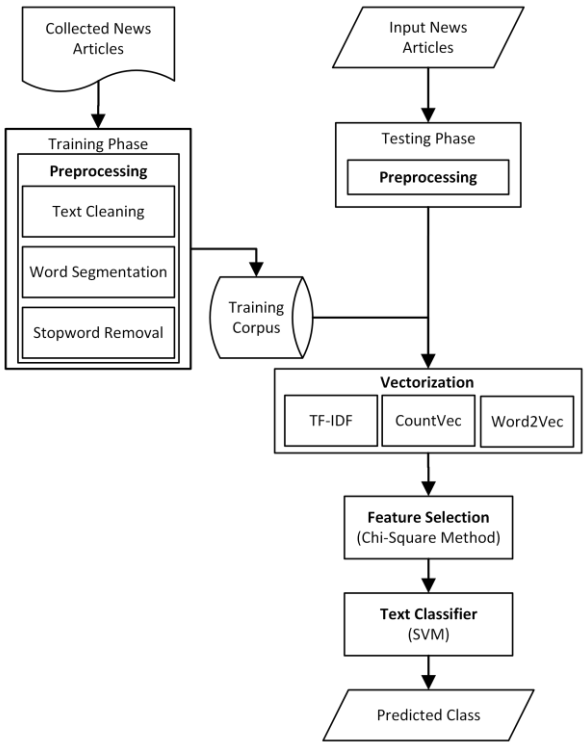

Figure 1. Proposed System Architecture

As shown in Figure 1, there are separate training and testing phases in the system. Firstly, the news articles are collected and applied to the training phase to work on preprocessing. Then, these training corpus are applied to the vectorizer to extract the features as well as to select features to be used. However, in the testing phase, new input news articles are accepted, and they are preprocessed the same as training articles. Later, they are passed through the vectorization and classifier to get the predicted class.

A. Preprocessing

Preprocessing is crucial to text mining methods because it influences the quality of the classification stage by determining which keywords are meaningful and should be retained while eliminating those that do not contribute to distinguishing between the documents. Obtaining the news's key feature from the documents and to improve the relevancy between words, documents, as well as categories, are the main objectives for the application of preprocessing. Text cleaning is the first stage and both Burmese and English numbers, symbols, and special characters from the Myanmar News dataset are carried out using regular expressions (for

example- ၂၀၂၂, ၊ , ။ , (,) , URLs, etc.). Then, all the English text which are embedded within the Myanmar sentences (eg- person, company, and movie names) are converted to lowercase to ensure uniformity in the processing of the text. The second stage of text classification is word segmentation. In languages where words are separated by spaces, like English, it is simple to break down sentences into individual words. However, languages such as Japanese, Chinese, Thai, Hindi, and Burmese do not have clear spaces between words. So, word segmentation is vital for the text preprocessing step. For this task, the "Pyidaungsu" Python library [5], which employs a technique called Conditional Random Field (CRF) prediction to tokenize sentences into words is utilized. As a final stage, the proposed system receives input words that have been previously segmented and undergoes the removal of Myanmar stop words such as စေ၊ သူ၊ ဒါ။ နိုင်၊ ဟုတ်၊ သည်, etc. About 390 Myanmar stop words are collected for pre-processing. Additionally, these words have very low discrimination value because, in every Myanmar document, they occur a lot of times. For this reason, they do not help in distinguishing between documents with contents that are about different topics.

Table 1. Example of Preprocessing

Input Text	ပူလီဆစ်သည် ၂၀၂၂ ခုနှစ်အတွင်း လက်ရွေးစင်အသင်းအတွက် ခုနစ်ပွဲကစား၊ ခြောက်ဂိုးသွင်းယူပေးပြီး (၂၀၂၂-၂၀၃၃) ကွန်ကာကပ်ဖ် Nations League ပြိုင်ပွဲ၌ အကောင်းဆုံးကစားသမားဆုကို ဆွတ်ခူးရရှိခဲ့သည်။
Cleaned Result	ပူလီဆစ်သည် ခုနှစ်အတွင်း လက်ရွေးစင်အသင်းအတွက် ခုနစ်ပွဲကစား ခြောက်ဂိုးသွင်းယူပေးပြီး ကွန်ကာကပ်ဖ် nations league ပြိုင်ပွဲ၌ အကောင်းဆုံးကစားသမားဆုကို ဆွတ်ခူးရရှိခဲ့သည်
Segmented Result	ပူလီဆစ်_သည်_ခုနှစ်_အတွင်း_လက်ရွေးစင်_အသင်း_အတွက်_ခုနစ်_ပွဲ_ကစား_ခြောက်_ဂိုးသွင်းယူ_ပေး_ပြီး_ကွန်_ကာကပ်ဖ်nationsleagueပြိုင်ပွဲ_၌_အကောင်းဆုံး_ကစားသမား_ဆု_ကို_ဆွတ်ခူး_ရရှိ ခဲ့_သည်
Remove Stop words	ပူလီဆစ်_ခုနှစ်_လက်ရွေးစင်_အသင်း_ခုနစ်_ပွဲ_ကစား_ခြောက်_ဂိုးသွင်းယူ_ကွန်_ကာကပ်ဖ်nationsleagueပြိုင်ပွဲ_အကောင်းဆုံး_ကစားသမား_ဆု_ဆွတ်ခူး

B. TF-IDF Vectorizer

The Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer is a widely used technique for translating text documents into numerical representations. Firstly, it measures term frequency, or TF, which is the total number of times a term appears in a document. Next, it performs the inverse document frequency (IDF) to reduce the weight of terms that appear frequently across documents. Finally, TF-IDF assigns higher weights to terms that are common within a text but rare across thecorpus. This suggests their significance in distinguishing between documents. TF-IDF Vectorizer helps with accurate news categorization in Myanmar by highlighting words and phrases that are distinctive of particular subjects or categories.

a) Term Frequency *tf(t,d),* is the frequency of term *t* in a document *d*,

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}, \quad (1)$$

where $f_{t,d}$ is the count of a term in a document, i.e., the number of times term *t* appears in a document *d*. The denominator shows the total number of terms in document *d* (This counts each occurrence of the same term separately).

b) Inverse document frequency *(IDF)* is a measure of how much information the word provides, i.e., if it is common or rare across all documents. It is the logarithmically scaled inverse fraction of the documents that contain the word (obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient):

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|} \quad (2)$$

Where:

-*N*: total number of documents in the corpus $N = |D|$
-$|\{d \in D : t \in d\}|$: number of documents where the term *t* appears (i.e., *tf(t,d) != 0*). If the term is not in the corpus, it will lead to a division-by-zero. So, it is common to adjust the numerator *1+N* and denominator to $1+|\{d \in D : t \in d\}|$.

c) Term Frequency – Inverse document frequency (TF-IDF)

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D) \quad (3)$$

A high weight in *tf–idf* is reached by a high term frequency and a low document frequency of that term in the whole collection of documents; the weights hence tend to filter out common terms. Since the ratio inside the *idf's* log function is always greater than or equal to 1, the value of *idf (and tf–idf)* is greater than or equal to 0. As a term appears in more documents, the ratio inside the logarithm approaches 1, bringing the *idf* and *tf–idf* closer to 0.

Since TF-IDF weights words based on relevance, one can use this technique to determine that the words with the highest relevance are the most important. This can be used to help classify articles more efficiently to certain categories.

C. Count Vectorizer

CountVectorizer is a simple and efficient technique for vectorizing text data. Every document is transformed into a vector that shows the total count number for each term that are present in the document. Although TFID Vectorizer considers a term's rarity, CountVectorizer just considers the frequency of occurrence when comparing texts. When it comes to text classification, CountVectorizer is able to identify phrases at the surface level, but it might not be able to identify the importance of distinguishing terms.

In CountVectorizer, the text is first tokenized, breaking it down into individual words or tokens. It builds a vocabulary of all unique words present in the corpus. Let's denote this vocabulary as *V*, containing *K* unique words. Once the vocabulary is established, each document is represented as a vector of word counts. The length of this vector is equal to the size of the vocabulary, i.e., *K*. For each document d_i, CountVectorizer constructs a *K*-dimensional vector v_i,

where each element $v_{i,j}$ represents the count of word *j* (from the vocabulary) in document *i.*

Mathematically, this can be represented as:

$$\mathbf{v}_i = [\text{count}(w_1, d_i), \text{count}(w_2, d_i), ..., \text{count}(w_K, d_i)] \quad (4)$$

Where:

- *count(w_j, d_i)* represents the number of occurrences of word w_j in document d_i.
- w_j represents the j^{th} word in the vocabulary *V*.
- *K* is the size of the vocabulary.

Since the majority of documents will only contain a subset of the vocabulary, the resulting feature matrix can be quite sparse. This means that there will be a lot of zero entries in the feature matrix. The CountVectorizer provides a straightforward method for converting text input into numerical feature representations by counting the instances of each word in the document. These vectors can then be used as input for machine learning algorithms such as Support Vector Machines (SVM) for classification tasks.

D. Word2Vec Vectorizer

Word2Vec is a neural network-based approach for vectorizing text data, which learns distributed representations of words in a continuous vector space. In the case of TFID and CountVectorizer, they produce sparse representations. However, this method is able to learn vector representations that maintain semantic and syntactic similarities, including word relationships by utilizing the context that surrounds each word. It can also be used to extract semantic connections and complex patterns from news articles. Thus, increasing the discriminating power of the classifier.

Continuous Bag of Words (CBOW) architecture [6] was used in this vectorizer. The goal is to use the context words to predict the target word within the predetermined window size. Word2Vec requires a large corpus of text data for training as well as word embeddings, and distributed representations are learned by Word2Vec in a continuous vector space. Each term in the lexicon is represented by a fixed-dimensional, dense vector.

The CBOW model is trained by maximizing the probability of predicting the target word given its context words. This is achieved using the negative log-likelihood (NLL) loss function. Once the CBOW model is trained, the learned weights of the neural network represent the word vectors. These vectors capture the semantic relationships between words in the corpus. The CBOW model maximizes the log probability of predicting the target word given its context words. This can be represented as:

$$Maximize \sum_{t=1}^{T} \log p(w_t | context(w_t)) \quad (5)$$

Where:

- *T* is the total number of words in the corpus.
- w_t is the target word.
- *context(w_t)* shows the context words of the target word w_t.
- *p(w_t / context(w_t))* is the conditional probability of observing the target word w t given its context words.

With the Word2Vec model, distributed representations of words are learned by training a neural network to predict target words given their context, or the other way around. These acquired word vectors represent the semantic connections among words and can be utilized as input features for machine learning algorithms such as SVM.

E. Feature Selection

In the feature selection stage, some of the features in the training set is collected. These features are typically the words presented in the training data. Feature selection is an important process as it helps training more easily and also it helps the classifier to see only a subset of the most important vocabulary. Moreover, applying feature selection aids in the enhancement of the classification accuracy when met with new unseen data. Using a lot of non-important noise features can generate more errors on the unseen data. Therefore, to specify the feature size, the proposed system utilizes the Chi-Square *(χ2)* method on all the categories. Using the two-way contingency table of a word *t* and a category *c*. i) *A* is the number of times *t* and *c* cooccur, ii) *B* is the number of times *t* occurs without *c*, iii) *C* is the number of times *c* occurs without *t*, iv) *D* is the number of times neither *c* nor *t* occurs, and vi) *N* is the total number of sentences − the word goodness measure is defined as follows:

$$x^2(t, c) = \frac{N \times (AD - CB)^2}{(A + C)(B + D)(A + B)(C + D)} \quad (6)$$

From equation (6), words that have a χ2 test score larger than 2.366 which indicates statistical significance at the 0.5 level are selected as features for respective categories [2].

To understand how many features are best for the task, the feature selection process is evaluated with the help of the SVM model. Features ranging from 3,000 to 7,102 are fed into the SVM model and accuracies are calculated. According to Figure 2, a feature count of about 5,000 represents the best among others. Therefore, for feature coverage of the system, 5,000 features are selected.

```
k_best: 3000, with the shape (6769, 3000) has an Average Accuracy: 0.8342456885022724
k_best: 3500, with the shape (6769, 3500) has an Average Accuracy: 0.8345416553400125
k_best: 4000, with the shape (6769, 4000) has an Average Accuracy: 0.8373481546014601
k_best: 4500, with the shape (6769, 4500) has an Average Accuracy: 0.8379391057238086
k_best: 5000, with the shape (6769, 5000) has an Average Accuracy: 0.8422229282048427
k_best: 5500, with the shape (6769, 5500) has an Average Accuracy: 0.838381909668432
k_best: 6000, with the shape (6769, 6000) has an Average Accuracy: 0.8373477179111793
k_best: 6500, with the shape (6769, 6500) has an Average Accuracy: 0.8389726424456402
k_best: 7102, with the shape (6769, 7102) has an Average Accuracy: 0.8336547373799238
```

Figure 2. Finding the best numbers of features using Chi Square

Both feature extraction and feature selection are critical steps in machine learning, enhancing model interpretability, generalization, and efficiency by reducing data complexity while preserving its crucial information.

F. Support Vector Machine Classifier

Support Vector Machine (SVM) is one of the supervised machine learning algorithms and it works greatly against classification tasks. It supports the application of high dimensional data for classification problems and SVM can be mainly used for credit risk analysis, text classification, extraction, and medical diagnosis. Moreover, for the repeated training set, it can provide the same hyperplane. Furthermore, SVM is good at generalization and the complexity of the classifiers is not dependent on the data dimensions but on the number of support vectors utilized.

SVM also has the benefits of protection against overfitting, and they can work well with sparse data.

Radial Basis Function (RBF) [7], one of *the SVM* models can also be used to classify data. This is an efficient method, as it maps the input data into a higher-dimensional space where it becomes linearly separable. Moreover, each training sample is given a weight, and this method works well for tasks involving complicated data patterns. This also enables it to categorize unidentified data based on how close they are to support vectors. When compared to linear kernel, whenever the data shows non-linear correlations, the RBF SVM model outperforms the Linear SVM model. Further, in applications such as text classification, where the data are naturally non-linear and have complex feature relationships, the RBF SVM model is the best choice.

For a dataset with text documents, each associated with a particular topic label, the RBF kernel can measure the similarity between two text documents in a high-dimensional feature space. The RBF kernel function is defined as:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \quad (7)$$

Where:

- x_i and x_j are feature vectors representing two text documents.

- $||x_i x_i - x_j x_j||$ represents the Euclidean distance between the two feature vectors.

- γ is a hyperparameter that controls the width of the Gaussian function. It determines the influence of each training example on the decision boundary.

The decision function for RBF SVM in topic classification is given by:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \exp\left(-\gamma \|x_i - x_j\|^2\right) + b \quad (8)$$

Where:

- *f(x)* is the class label for document *x*.

- $\alpha_i \alpha_i$ are Lagrange multipliers that are computed during the learning phase.

- $y_i y_i$ are the class labels of training documents *xi*.

- $||xx - x_i x_i||$ symbolizes input document *x* from support vectors *xi* using Euclidean norm.

- *b* is an intercept also referred to as a bias term.

I.The function for RBF SVM in topic classification is similar to the standard SVM function and is given by:

$$Minimize \ \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \exp\left(-\gamma \|x_i - x_j\|^2\right) - \sum_{i=1}^{N} \alpha_i$$
$$Minimize \ \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \exp\left(-\gamma \|x_i - x_j\|^2\right) - \sum_{i=1}^{N} \alpha_i$$

$$Subject\ to: 0 \leq \alpha_i \geq C\ for\ all\ i, and\ \sum_{i=1}^{N} \alpha_i y_i = 0$$
$$Subject\ to: 0 \leq \alpha_i \geq C\ for\ all\ i, and\ \sum_{i=1}^{N} \alpha_i y_i = 0 \quad (9)$$

Where:

- *C* is a regularization parameter that controls the tradeoff between the maximum and minimum margin in the error classification.

- The first and second terms represent margin and classification error respectively.

- The multipliers $\alpha_i \alpha_i$ are constrained to be non-negative and their sum should equal zero.

This approach for classification uses RBF kernel to measure similarity between text through which it learns the best decision boundary to classify into different topics.

IV. EXPERIMENTAL WORK

A. *Data utilized for experiment*

To conduct the experiment, several Myanmar news websites' [8, 9, 10, 11] data such as speech transcription and textual news are collected. Since each category has 180 articles, the dataset with four categories consists of 720 articles. A typical document from the dataset contains an average of six sentences per document. The challenge is that when articles are taken as input, they may contain mixed topics. It may become difficult for the model to know what belonged where. Data transformation of a whole article at the sentence level resolves the class imbalances and contributes to more precise error detection, and increasing the overall classification rate. Thus, all articles are broken down into sentences and then the sentences for each category are divided into 80-20 train-test splitting. The training set has 6,769 news sentences while the test set has 1,695 news sentences and they are created using "scikit-learn" library.

Table 2. Data used for an experiment

Data (Sentences)	Business	Crime	Entertainment	Politics
Training	1,497	1,436	2,084	1,752
Testing	375	359	522	439
Total	1,872	1,795	2,606	2,191

B. *Performance Measures and Results*

To evaluate the test set performance, metrics like F1-score, recall and precision can be calculated. The calculation on each metric can be carried out when the news sentences are assigned to a particular category based on the content. The precision of the test set for a category is the ratio of correctly classified documents counts to documents labeled counts by the system. Recall is the ratio of the correctly classified documents to the number of documents of that category in training data. The F1 score can be stated as a weighted average of recall and precision. Finally, the accuracy measures the overall correctness of the system.

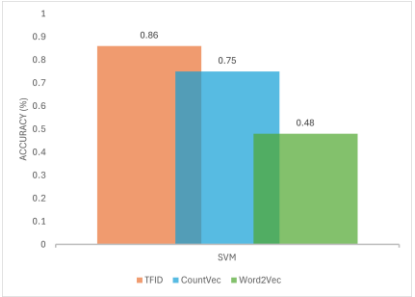

Figure 3. Experimental Results of Vectorizers on SVM Model

From Figure 3, it is obvious that TFID and CountVec perform better than Word2Vec. Depending on the nature of the dataset, preprocessing methods, and classification, the Word2Vec vectorization technique (which learns distributed representations of words by training a neural network)

Table 3. Experimental Results of Word2Vec, CountVec and TF-IDF Vectorizers on categories

Category	*Word2Vec*			*CountVec*			*TF-IDF*		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Business	0.67	0.41	0.5	0.88	0.71	0.79	0.91	0.83	**0.87**
Crime	0.83	0.05	0.1	0.86	0.65	0.74	0.89	0.84	**0.86**
Entertainment	0.46	0.7	0.55	0.65	0.94	0.77	0.83	0.94	**0.88**
Politics	0.44	0.64	0.52	0.83	0.67	0.74	0.85	0.83	**0.84**

doesn't scale well. CountVec has an accuracy of 75%. However, from the above figure, it is safe to conclude that the TFID vectorizer with 86% accuracy is the best vectorizer for Topic Classification on Myanmar News Dataset.

From the experimental results of Table 3, both Count Vectorizer and TFID Vectorizer show high precision and recall, resulting in high F1-scores for categories. Word2Vec has lower precision and recall compared to the other two vectorizers. In addition, regarding precision and recall, TFID Vectorizer outperforms CountVectorizer in all categories as well. TFID Vectorizer and CountVectorizer are identical in terms of recall for the "Entertainment" category. Moreover, the Word2Vec has the lowest recall and F1 score in the 'Crime' category.

According to Table 3, TFID beats the CountVectorizer and Word2Vec Vectorizers in every area with great recall, precision, and F1-score. This is because TFID Vectorizer is able to capture each category's unique features. Furthermore, the CountVectorizer does well, but the TFID does the best in most categories. However, whenever new data comes and there are no matched features in the vectorizer, the model will not be able to correctly classify the data to right category.

TFID is capable of focusing on terms that are more useful for categorization. Because it considers the significance of each phrase in the document as well as the full corpus. However, CountVectorizer counts the occurrence of terms in the document without taking into account their relevance in the corpus. So, it is not able to capture the significance of terms as effectively as TFID. Despite Word2Vec being a strong embedding technique, it is also not able to accurately classify texts. Because Word2Vec represents words as a dense vectors based on similarity but does not directly consider the frequency of terms. Based on the findings, the TFID vectorizer is generally very effective across all categories in the sense that it extracts the right features from the articles for classification. Therefore, among the three vectorizers, it becomes evident that the TFID Vectorizer is the most appropriate for the topic classification tasks.

V. CONCLUSION

In this research paper, through many experimentations and evaluations using an RBF SVM model, the proposed system has proven the effectiveness of TFID Vectorizer as the best option for Myanmar news topic classification. The comparison research proved that it has performed better than Word2Vec vectorizer and CountVectorizer across metrics: F1-score and its precision and recall across all categories.

In terms of the analysis of the dataset, TFID Vectorizer has an outstanding performance in determining the relevance of words in documents. The TFID Vectorizer is capable of weighing the terms according to their relevance in the given corpus to ensure more precise classification results. While the count vectorizer performs reasonably well, the TFID Vectorizer was slightly better in terms of precision, recall and overall F1-score.

To sum up, the findings highlight the importance of the proper selection of vectorization methods in NLP tasks, particularly when working with languages other than English, like Myanmar. This research contributes to the existing body of knowledge in natural language processing (NLP) especially because it demonstrated the comparative research on different vectorization techniques and provides proof of how TFID Vectorizer is more appropriate for the classification of news from Myanmar. Howerver, this study has limitations such as the dataset size, sentence granularity and challenges with mixed-topic articles. Future work could continue to investigate the deep down analysis of why some vectorization techniques perform poorly and on which kind of data and strategies to overcome. Additionally, new categories can be added to help evaluate the scalability of the vectorization methods for more complex classification tasks.

REFERENCES

[1] T. T. Zaw and K. M. Soe, "Myanmar Text Classifier Using Genetic Algorithm," *National Journal of Parallel and Soft Computing*, pp. Volume 01, Issue 01, 2019.

[2] K. T. Nwet, A. H. Khine and K. M. Soe, "Automatic Myanmar News Classification," in *Fifteenth International Conference On Computer Applications (ICCCA2017)*, Yangon, 2017.

[3] H. Aung and W. P. Pa, "Analysis of Word Vector Representation Techniques with Machine-Learning Classifiers for Sentiment Analysis of Public Facebook Page's Comments in Myanmar Text," in *2020 IEEE Conference on Computer Applications (ICCA)*, Yangon, 2020.

[4] U. Gumińska, A. Poniszewska-Maranda and J. Ochelska-Mierzejewska, "Systematic Comparison of Vectorization Methods in Classification Context," *Applied Sciences*, pp. 5119-5133, 2022.

[5] K. H. San, "Pyidaungsu library for Myanmar language," 2020. [Online]. Available: https://github.com/kaunghtetsan275/pyidaungsu.

[6] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *Proceedings of Workshop at ICLR*, 2013.

[7] C.-w. Hsu, C.-c. Chang and C.-J. Lin, "A Practical Guide to Support Vector Classification," 2009.

[8] Eleven, "Eleven News Data," [Online]. Available: https://news-eleven.com/news.

[9] BBC, "BBC Myanmar News Data," [Online]. Available: https://www.bbc.com/burmese.

[10] 7daydaily, "7 Day Daily News Data," [Online]. Available: http://7daydaily.com/story.

[11] VOA, "VOA Myanmar News Data," [Online]. Available: https://burmese.voanews.com/.