Volume 11. No 3, September 2024
e-ISSN: 2527-9572 / ISSN: 2354-8924

Comparison of Feature Extraction Methods for Conducting Sentiment Classification in Ternate Malay Language using Machine Learning Approaches

***Satria Dwi Surya**
Teknik Informatika
Universitas Amikom (Magister Teknik Infromatika) Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta, 55283, Indonesia
satriadwisurya@students.amikom.ac.id

Ema Utami
Teknik Informatika
Universitas Amikom (Magister Teknik Infromatika) Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta, 55283, Indonesia
ema.u@amikom.ac.id

Ainul Yaqin
Teknik Informatika
Universitas Amikom (Magister Teknik Informatika) Jl. Ring Road Utara, Condong Catur,Sleman, Yogyakarta, 55283, Indonesia
ainulyaqin@amikom.ac.id

***Abstract* – Local people in Ternate, North Maluku, often use local languages to communicate on social media. This poses a challenge for newcomers to understand the implied meaning and emotions of the messages conveyed through social media. This research aims to develop a natural language processing (NLP)-based emotion classification method that can be applied to Ternate Malay text datasets. The application of NLP is expected to improve the accuracy of emotion detection and classification in the text. The research was conducted by applying and comparing the performance of several classification models trained using Ternate Malay text datasets. The models used include SVM (Support Vector Machine), K-Nearest Neighbors (KNN) Random Forest, Decision Tree and Logistic Regression. Each model is applied using BoW (Bag-of-Words) and Word2Vec vectorization representations. The evaluation results show that the BoW+SVM model provides the highest performance with 77% accuracy, followed by BoW+Random Forest (75%) and BoW+Logistic Regression (73%). Thus it can be concluded that NLP can be applied to the Ternate Malay language dataset to classify emotions based on text.**

Keywords: Text Classification, Emotion, Artificial Intelligence, Machine Learning, Ternate Malay Language

I. INTRODUCTION

Emotion is a psychological phenomenon that reflects various mental states and plays an important role in shaping individual human characteristics. Sentiment analysis is a field of study used to analyze the opinions, attitudes, and emotions of a person in a text document towards products, services, and other things. It helps in understanding the perceptions and reactions of individuals to various aspects [1] [2] [3]. According to the explanation [4], emotion serves as a response to individuals or events, and plays a significant role in daily experiences in socializing. The range of emotions includes three main types of feelings, namely Positive, Neutral, and Negative, which have been identified and described [4]. In this context, research often identifies specific emotional categories, but not limited to anger, happiness, sadness, and fear, as described by[5]. Emotion assessment has become a prominent focus in various scientific disciplines, covering cognitive science, psychology, and even extending to the realm of social media. This draws the attention of many computer scientists[6], as in their research[7]. Exploration of societal trends, especially those related to psychological phenomena [8], has proven to be very valuable [9][10]. Social media has become a major channel for individuals to express emotions[11], various opinions, and explore aspects of daily life, as observed by[1].

With the development of information technology, there is an increase in people's use of information media, especially in the context of social media [12]. Local people in Ternate, North Maluku, often use local languages to communicate on social media [13]. This is a challenge for migrants to understand the meaning and emotions implied by the messages conveyed [14][15]. Natural Language Processing (NLP) and Machine Learning can help overcome this challenge by developing artificial intelligence models to analyze text in local languages and classify emotions. [16]. A survey conducted between 2014 and 2018 revealed that six out of ten research papers focused on emotion classification in textual content, as reported by [17] In the field of text classification, many machine learning algorithms have been used. Similarly for feature extraction, some techniques have become popular, including Bag Of Words (BoW) and Word2Vec [18], techniques, as stated by [19]. With this contextual foundation in place, the researchers embarked on an effort to answer specific questions. These questions centered on the feasibility of applying NLP approaches to the Ternate

Malay language and subsequent performance metrics, including Accuracy, Recall, Precision and F1-Score, of the machine learning models used. This research aims to develop an NLP-based emotion classification method that can be applied to Ternate Malay text datasets. NLP provides techniques to process human language data and extract meaningful patterns. Machine learning algorithms can then utilize these patterns to train classification models. The application of NLP and machine learning is expected to improve the accuracy of detecting and categorizing emotions from text written in local languages [20]. By comparing the performance of NLP and machine learning models trained on a Ternate Malay text dataset, this research aims to demonstrate the feasibility and values of these techniques for local language emotion analysis. The researchers used a public dataset from kaggle.com (emosi_melayu_ternate) as a contribution and experimental material for this research [21].

II. METHOD

A. Dataset

In this study, an experiment was conducted using a sample of the Ternate Malay dataset taken from a public source, specifically from Kaggle.com (emosi_melayu_ternate). This dataset contains text data that conveys various emotional expressions. Each text in the dataset is equipped with an emotion label, which consists of nine categories, namely anticipation, joy, disgust, anger, blame, trust, sadness, fear, and surprise.

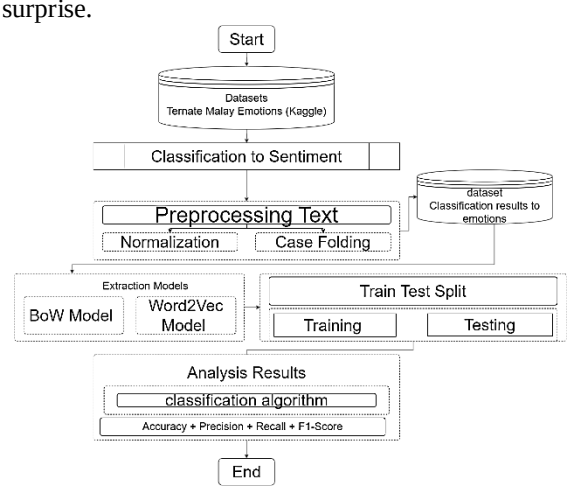

Figure 2. Stages of Data Analyisis

B. Emotion Classification

This phase involves classifying emotions in the texts into Negative, Positive, or Neutral categories based on the labels associated with each text. By linking emotions like anticipation, joy, disgust, anger, etc. to these broader categories, a deeper understanding of the emotional nuances within the Ternate Malay texts can be gained. Through analysis, the various emotion forms will be associated with the appropriate emotion category. This emotion classification process will provide a more holistic overview of how complex emotions translate to simpler forms. Once classification is complete, the labeled emotions will be integrated into the dataset to supplement the information for each text. Overall, this emotion categorization will facilitate better comprehension of the emotion types and patterns in the Ternate Malay corpus.

C. Feature Extraction Model

In this experimental phase, three different feature extraction methods will be tested, Bag-Of-Words (BoW), and Word2Vec. These methods will be applied to the processed dataset. Each feature extraction method employs a unique approach to convert text into numerical representations suitable for further analysis and modeling.

a. Bag-of-Words (BoW)

The BoW method transforms each text into a vector based on the frequency of word occurrences within that text. Each word is treated as an independent feature, and the text is represented as a vector with dimensions corresponding to the size of the specified vocabulary [22]. The following formula used to perform Vector representation using BoW (1):

$$BoW(d) = [n_1, n_2, \ldots., nm] \quad (1)$$

Where $n_1$represents the number of occurrences of the i -th word from the vocabulary in document d, and m is the count of unique words in the vocabulary.

b. Word2Vec

The Word2Vec method focuses on recognizing the contextual meaning of words in the text. It generates vector representations that reflect the semantic relationships between words in the text. The Word2Vec model learns the contextual similarity of words and represents them as vectors in a semantic space [23]. The following formula used to perform Vector representation using Word2Vec (2), (3) and (4):

$$h = W_{in} . x \qquad (2)$$

where in the input layer, x represents the one-hot encoding input vector representating the target word, and W_{in} denotes the weight matrix connecting the input to the hidden layer.

In the hidden layer, h represents the hidden representation vector. Moving th the output layer W_{out} denotes the weight matrix connecting the hidden layer to the output point, and $\hat{y}$ signifies the probability distribution of words in the vocabulary.

$$\hat{y} = softmax(W_{out} . h) \qquad (3)$$

The softmax function is defined as softmax:

$$softmax(z)_i = x_{y^2} \frac{e^z i}{\sum_{j=1}^{v} e^z j} \quad (4)$$

Where is z the input vector, z_i is the i -th element of that vector, and V is the vocabulary size.

D. Comparison of Machine Learning Algorithms

A 80/20 train-test split is used with stratification ("stratify=y") to reduce bias by fairly representing classes in both splits. Random state is set ("random_state=100") for reproducibility and consistency between experiments. In summary, a stratified train/test configuration allows examining algorithm performance across diverse contexts while random state setting ensures reliable, reproducible comparisons supporting robust analysis. This research employs various machine learning algorithms to facilitate comparative analysis. Specifically, several classification algorithms, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression (LR), and Random Forest (RF), are utilized for this purpose. [22]. The following is a description of the formula for the classification algorithm:

a. Support Vector Machine (SVM) (5)(6) and (7):
SVM is a classification algorithm that utlillizes a decision function defined as:

$$f(X) = sign\left(\sum_{i=1}^{N} \alpha_i y_i K(X_i, X) + b\right) (5)$$

Here , α_i represents the weights obtained during training, y_i is the class label of the training example, $K(X_i, X)$ is the kernel function measuring the similarity between vectors X_i and X in feature space, and b is the bias or offest. The liner kernel function.

$$(K(X_i, X) = X_i . X) \quad (6)$$

is one of the kernel options, determining how similar two vectors are in feature space. The hyperplane equation, which represents the decision boundary two classes, is given by:

$$\sum_{i=1}^{N} \alpha_i \, y_i K(X_i, X) + b = 0 \quad (7)$$

b. K-Nearset Neighbors (8), (9):

$$\hat{y}(X) = argmax_y \sum_{i}^{k} I(y_i = y) \, (8)$$

The predicted class $\hat{y}(X)$ for input X is determined as the class $\hat{y}$ that maximizes the sum, where $I(y_i = y)$ is an indicator function tha outputs 1 if the class of the i -th neighbor (y_i) is equal to y and 0 otherwise. In other words, $\hat{y}(X)$ is the predicted class for X based on the class labels of its K-Nearest neighbors. The Euclidean distance $d(P,Q)$ between two points $P$and Q in an n – dimensional feature space is expressed as:

$$d(P,Q) = \sqrt{\sum_{i=1}^{n} (P_{i,} - Q_i)^2} \quad (9)$$

Where P_i and Q_i are the i-th components of vectors P and Q.

c. Logistic Regression (10), (11) , (12), (13) and (14):
In the linear combination of feature like the following equation:

$$zC_k = bC_k + \sum_{i=1}^{n} WC_{k,} iX_i \quad (10)$$

Represents the result of combining features for class C_k, where bC_k is the bias term, $WC_{k,}i$ is the value of feature i for a given sample. Applying the softmax function to these linear combinations yields the class probabilities:

$$P(C_k) = \frac{e^{zC_k}}{\sum_{j=1}^{k} e^{zj}} \quad (11)$$

Where e^{zC_k} is the exponential of the lienar combination results for class C_k and $\sum_{j=1}^{k} e^{zj}$ is the sum of exponentials of the lienar combination results for all classes C_j. For prediction, the class with the highest probability is assigned as the predicted class:

$$Predicted\ Class = argmax C_k P(C_k) \quad (12)$$

The log-likelihood or cross-entropy cost function is then used to measure the model's perfomance

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} I(y^{(i)} = (C_k)\ log\ P(C_k^{(i)}) \, (13)$$

Where $I(y^{(i)} = (C_k)$ is the indicator function and $logP\left(C_k^{(i)}\right)$ is the logarithm of the predicted probability for class C_k on the i-th sample. The gradient descent update rule for the weights is given by :

$$wC_k, i := wC_k, i - \alpha \frac{\partial J(\theta)}{\partial \omega C_{k'i}} \quad (14)$$

Where wC_k, i is the weight for feature i in class C_k, α is the learning rate, and $\alpha \frac{\partial J(\theta)}{\partial \omega C_{k'i}}$ is the partial derivate of the cost function with respect to the weight wC_k, i

d. Decision Tree (15), and (16).
Entropy (S) is calculated using the formula :

$$Entropy(S) = \sum_{i=l}^{n} -p_i * log_2 p_i \quad (15)$$

Where, S represents entropy, measuring the level of uncertainty or disorder in a dataset. Here, n is the number of classes or catagories in the dataset, $p_i$is the proportion of samples belonging to class i (from class 1 to n), and log_2 denotes the logarithm base 2. Information gain:

$$Gain\ (A,S) = S - \sum_{i=1}^{n} \frac{|S_i|}{|S_i|} x\ Entropy\ S_i \quad (16)$$

Representing the information gain by splitting the dataset S based on attribute A, $|S_i|$ is the number of instances in subset S_i and n is the number of distinct values for attribute A.

e. Random Forest (17):
Boostrap sampling involes randomly sampling, with replacment, from the original dataset to create multiple boostrapped samples. Random feature selection is implemented by randomly selecting a subset of features for each decision tree during the training process. This aids decorrating the trees. Decision tree training is accomplished by growing decision trees using the bootsrapped samples and random feature subsets. Each tree is trained to predict class labels for the given task. Voting for classification, a common strategy is "soft voting" where the class with the highest probability (or the sum of class probabilites) across all trees is chosen as the final predicted class.

$$\hat{y} = arg\ max_k \sum_{i-1}^{N} P_k^{(i)} \quad (17)$$

Here, $\hat{y}$ represents the predicted class, N is the number of trees, and $P_k^{(i)}$ is the probability assigned k by the i-th tree.

The final stage in carrying out the evaluation is to test each performance of each classification algorithm using a classification report [24]. The following is the equaiton of a classification report (18), (19), (20) and (21).

Presision (P) is calculated using the formula:

$$p = \frac{TP}{TP + FP} \quad (18)$$

Where TP is the number of True Positives (corectly predicted positive instance) and FP is the number of False Positives (incorectly predicted positive instances).

recall (R), also know as Sensitivity or True Positive Rate (TPR) is calculated as:

$$R = \frac{TP}{TP + FN} \quad (19)$$

Where FN is the number of False Negatives (positive instances incorrectly predicted as negative).

F1-Score is computed as the harmonic mean of precision and recall:

$$F1 = \frac{2.P.R}{P + R} \quad (20)$$

Accuracy is calculated using the formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (21)$$

Where TN is the number of True negatives (correctly predicted negative instances), and FP is the number of false postives. These metrics provide a comprehensive evaluation of a classification model's perfomance for each in a multicalss classification scenario.

The Weighted-Average denotes an independent calculation of the mean value for each class, wherein the weighting is determined by the size of the dataset for each class. The formula for this calculation can be expressed as follows[25]:

Weighted Precision (22):

$$\text{W. Precision} = \frac{\sum w_i . P_i}{\sum_i w_i} \quad (22)$$

Where w_i is the weight assigned to class i, and P_i is the precision for class i.

Weighted Recall (23)

$$\text{W. Recall} = \frac{\sum w_i . R_i}{\sum_i w_i} \quad (23)$$

Where R_i is the recall for class i

Weighted F1-Score (24):

$$\text{W. F1-Score} = \frac{\sum w_i . F1_i}{\sum_i w_i} \quad (24)$$

Where $F1_i$ is the F1-Score for class i

III. RESULTS AND DISCUSSION

A. DataFrame

The dataset utilized in this research is secondary data obtained by the researcher from reliable sources, specifically chosen to support the undertaken study. The data has undergone a process of selection and adjustment based on the needs of the analysis and to address the research questions formulated beforehand. The dataset is ensured to be relevant and capable of representing the issues intended to be explored more deeply through this study.Here is the dataframe before undergoing the pre-processing stage based on Table 1.

Table 1. DataFrame Before Preprocessing Stage

label	teks
antisipasi	kita batabung dah,..jang sampe nanti so mo kaweng kong trada doi lagi,..akang bakuli di tambang suda
antisipasi	kita so siap forok deng loyang ni..jaga jaga sa,..jang sampe nanti di puncak tong butuh lagi,.bagemana..gaga?
antisipasi	e haiwan, polisi ada batilang sana jang sampe ngana dapa tilang, pake ngana pe helm tu
antisipasi	kita toh tidor musti pasang alarm ka lao tarada pasang akan tarada tabangun
antisipasi	e lebae ngana cari di toko juragan kamuka jang sampe toh di pasar tarada

The overall dataset needs to undergo text pre-processing to be ready for further analysis in emotion classification. Essential preprocessing steps include removing punctuation, word normalization, and other necessary adjustments.

Based on the distribution of the number of characters in the Ternate Malay emotion dataset, it can be observed as shown in Figure 3.

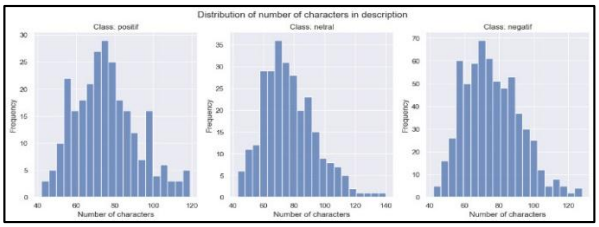

Figure 3. Distribution of Character Counts in the DataFrame

The common pattern observed from the image is that the distribution of the number of characters for all emotion classes follows a similar shape, resembling a bell curve. This indicates that the number of characters in emotional texts within the dataset has a normal distribution. The noticeable differences between emotion classes lie in their mean and variance values. The positive emotion class has a higher average number of characters compared to the negative emotion class. This suggests that positive emotion texts tend to be longer than negative emotion texts. Possible causes for the differences between emotion classes in terms of the number of characters could be attributed to various factors, including:

1. The fact that positive emotions are expressed more frequently than negative emotions.
2. The higher complexity of positive emotions compared to negative emotions.
3. The tendency of individuals to provide more detailed texts for positive emotions compared to negative emotions.

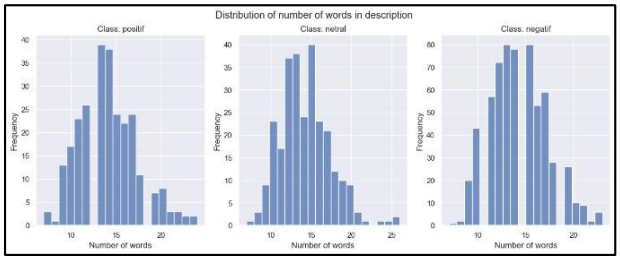

Figure 4. Distribution of Character Counts in the DataFrame

The Ternate Malay dataset shows variation in word usage across emotion classes. Positive emotions tend to use fewer words than neutral, while negative uses the most. Specifically, positive texts often have very few (as low as 10) words, indicating "easy to understand" language. Neutral texts range from 15-40 words, suggesting more "meaningful" and informative language. Finally, negative texts use up to 80 words, meaning the language may be "slow or difficult to understand." In summary, positive texts are simple, neutral informative, and negative complex in terms of word count and language. This distribution reveals differences in how the emotion classes utilize language, which can help develop a more accurate emotion model for Ternate Malay.

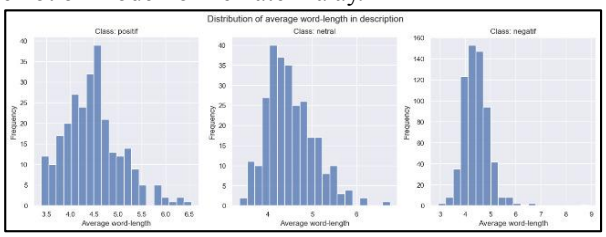

Figure 5. Distribution of Character Counts in the DataFrame

This pattern depicts the distribution of the average word length for all emotion classes in the Ternate Malay dataset. Consequently, this pattern can aid in identifying substantial differences between different categories within the texts.

Example:

1. In the description of the "negative" class, verbs with negative meanings are more frequently used, such as "tara bae = not good," "badaki = dirty," and "pastiu = bored."
2. In the "positive" class, verbs with positive meanings are more frequently used, such as "bae = good," "gaga = keren," and "danke = thank you.

The pattern suggests negative class texts have both advantageous and disadvantageous factors for those individuals, while positive class texts have both disadvantageous and advantageous factors. It also indicates a strong relationship between adjacent texts, possibly reflecting connections like learning ability or academic assessments. The high verb frequency suggests close linkage between the two texts in these cases. In summary, negative texts show mixed benefits, positive mixed drawbacks, and high verbs demonstrate substantial text interrelation likely along dimensions like capability or performance.

B. Emotion Classification

The pre-processing stage refines and organizes raw unstructured data that often contains noise like punctuation and irrelevant sentences. A key step is text folding - transforming text to lowercase and removing punctuation marks, HTML tags, and numbers to clean the data. Next, text normalization enhances coherence and expands abbreviated structures. Finally, a training model of previously classified text documents is used to train the machine learning classifier that will categorize new texts. Overall, pre-processing cleans and structures the raw textual data to prepare it for the machine learning algorithm. [26]. The training model comprises two sets: the first set collects 1173 labeled texts categorized into 3 categories such as neutral, negative, and positive. Here are the results from the preprocessing and classification stages based on Table 2.

Table 2. Result of Case Folding & Text Normalization

label	teks	emosi
antisipasi	kita batabung dah jang sampe nanti so mo kaweng kong trada doi lagi akang bakuli di tambang suda	netral
antisipasi	kita so siap forok deng loyang ni jaga jaga sa jang sampe nanti di puncak tong butuh lagi bagemana gaga	netral
antisipasi	e haiwan polisi ada batilang sana jang sampe ngana dapa tilang pake ngana pe helm tu	netral
antisipasi	kita toh tidor musti pasang alarm ka lao tarada pasang akan tarada tabangun	netral
antisipasi	e lebae ngana cari di toko juragan kamuka jang sampe toh di pasar tarada	netral

The distribution of classified emotions based on labels in the dataset can be seen in Figure 4.

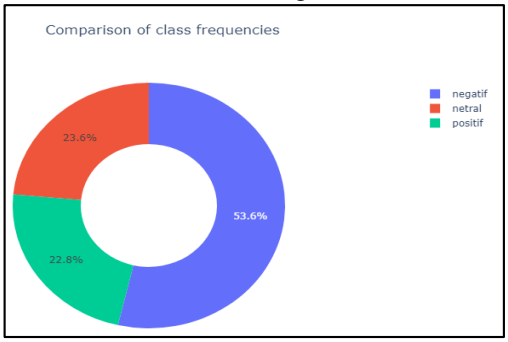

Figure 6. The Number of Each Emotion After Going Through the Classification Stage

Neutral emotion has the most data with 275 cases, indicating texts with neutral expressions dominate. Negative emotion is second with 626 cases, nearly 2.5 times the positive emotion data, suggesting

more negative emotion texts. Positive emotion has the least data with 266 cases, but still a significant number to represent positive emotions for machine learning. Overall there is an class imbalance, with neutral emotion having the most data, followed by negative and then positive emotion having the least cases.

C. Feature Extraction Model Technique

a. Bag-of-Words (BoW)

The following is pseudocode for the vectorization process using the Bag-of-Words (BoW) method in python programming language based on Table 3.

Table 3. Text to Bag of Words Model Pseudocode

Pesudocode (Bow Model Vectorization Technique)
Load dataset from 'dataset/dataset_hasil_klasifikasi.csv' into df texts = df['teks'][:5] vocabulary = empty set for each text in texts: words = convert text to lowercase and split into words add words to vocabulary word_counts = empty list for each text in texts: words = convert text to lowercase and split into words word_count = count occurrences of each word add word_count to word_counts bow_representation = empty list for each word_count in word_counts: vector = empty list for each word in vocabulary: count = get count of word from word_count add count to vector add vector to bow_representation bow_array = convert bow_representation to NumPy array bow_df = create DataFrame with columns 'Teks' and 'BoW_Vector' set display option 'max_colwidth' to None set display option 'expand_frame_repr' to False print bow_df

b. Word2Vec

Table 4. Text to Word2Vec Model Pseudocode

Pseodocode (Calculating the average vocabulary vector)
Function sent_vector(sentence, W2V): vecs = Empty List For each word in word_tokenize(sentence): Add W2V[word.lower()] to vecs Calculate sent_vec = Mean of vecs along axis 0 Return sent_vec
Pseodocode (perform vector normalization on the vocabulary)
Function norm_sent_vector(sentence, W2V): vecs = Empty List For each word in word_tokenize(sentence): Try: vec = W2V[word.lower()] Add vec to vecs Catch KeyError: Continue to the next word norm_vecs = Empty List For each vec in vecs: If ‖vec‖ > 0: Add vec / ‖vec‖ to norm_vecs Calculate sent_vec = Mean of norm_vecs along axis 0 Return sent_vec

The following is the pseudocode for the vectorization process using the Word2Vec method using the python programming language based on Table 4.

D. Comparison of Feature Extraction Models and Machine Learning

The following is the performance test results for each combination of feature extraction models and classification algorithms using a classification report matrix summarized based on the findings from Table 5.

Table 5. Summary of Performance Results for Algorithm and Feature Extraction Combinations

Algorithm	feature extraction	Train Test Split (Test_size =0.2, Stratify =y, random_state =100)				
		Weighted Avg			RandomSearchCV	
		Precision	Recall	F1-Score	Train Score	Test Score
SVM	BoW	0.76	0.77	0.76	1.0	0.7660
	Word2Vec	0.67	0.66	0.66	1.0	0.6638
KNN	BoW	0.66	0.67	0.65	0.7409	0.6723
	Word2Vec	0.70	0.70	0.69	0.7548	0.7021
LR	BoW	0.72	0.73	0.72	0.9947	0.7319
	Word2Vec	0.66	0.63	0.56	0.6354	0.6298
DT	BoW	0.67	0.67	0.66	0.8337	0.6723
	Word2Vec	0.52	0.53	0.53	0.9787	0.5319
RF	BoW	0.79	0.75	0.73	1.0	0.7532
	Word2Vec	0.69	0.68	0.64	1.0	0.6766

Overall, the combination of SVM algorithm and Bag of Words (BoW) feature extraction produces the most optimal performance compared to other combinations. This is shown by the highest values for precision, recall, F1-score, and accuracy, which are 0.76, 0.77, 0.76, and 0.7660, respectively. For the KNN algorithm, the BoW combination also outperforms Word2Vec, with a difference in F1-Score of about 2% and accuracy of about 3%. The Decision Tree and Logistic Regression algorithms also show the same pattern, where the BoW combination outperforms Word2Vec in terms of F1-Score and accuracy. Only in the Random Forest algorithm, the combination with Word2Vec has higher F1-Score and accuracy values, although the difference is not significant. In conclusion, for the case of emotion text classification in Ternate Malay, Bag of Words feature extraction is more suitable for most machine learning algorithms compared to Word2Vec, except for Random Forest which tends to be more suitable for Word2Vec.

The potential reasons for the lower performance of the Word2Vec model compared to Bag of Words in this Ternate Malay dataset can be analyzed as follows:

1. Word2Vec, as a model that relies on contextual word learning, seems to be less than optimal in mapping the semantic relations of words in Ternate Malay. This can be seen from the lower

precision, recall, F1-score, and accuracy values, which indicate the difficulty of the model in understanding the contextual meaning of words. In contrast, BoW does not require contextual understanding because it only calculates the frequency of word occurrence.

2. The quality and quantity of training data used may not be sufficient for Word2Vec to learn vector mapping of words in Ternate Malay effectively. A large amount of training data is required for Word2Vec to accurately form semantic relationships between words.
3. The limited computational resources used in training the Word2Vec model may also lead to sub-optimal results. Word2Vec learning requires significant resources to produce an accurate word vector representation.

Based on the theory of similar research with different languages by [20], there are several techniques adopted and optimized by this research such as optimizing the best parameters automatically using the randomSearchCV library in Python, where the library will provide the best parameters in each algorithm where the research [20] still uses manual methods to find the best parameters. then based on datasets that have unbalanced classes, this research also utilizes weighted averages to help give appropriate weights to each class so that the model evaluation reflects as far as possible the overall performance on all classes.

Therefore, it can be said that the Word2Vec model has not maximally understood the meaning of words in the context of Ternate Malay language in this study. Further refinements are needed to improve its contextual capabilities.

IV. CONCLUSION

Based on the results of the previous analysis and discussion, several conclusions can be drawn. The Natural Language Processing approach has proven to be applicable to Ternate Malay for text-based emotion classification. The combination of the SVM algorithm and Bag of Words (BoW) feature extraction yielded the most optimal performance with precision of 0.76, recall of 0.77, F1-score of 0.76, and accuracy of 0.77. In general, the Bag of Words feature extraction is more suitable for the KNN, Decision Tree, Logistic Regression, and SVM algorithms compared to Word2Vec in the case of emotion classification in Ternate Malay. The Word2Vec model has not optimally understood the contextual meaning of words in Ternate Malay, as indicated by its low metric values. Further refinement of the model with a larger training dataset is needed. There is an imbalance in the distribution of emotion label data in the dataset that needs to be addressed with data balancing techniques to improve model performance.

V. ACKNOWLEDGMENTS

The author expresses their gratitude to Professor Ema Utami, S.Si., M.Kom. and Mr. Ainul Yaqin M.Kom. for their invaluable support and guidance in completing this research. Professor Ema's input and motivation have improved the research process and prepared the report. Mr. Ainul's guidance in data analysis and drawing conclusions has been instrumental in the research's completion. The author expresses their deep appreciation for the time and effort devoted by these individuals, and wishes Allah Subhanahu wa ta'ala always bestow His blessings and grace upon them.

REFERENCE

[1] R. D. Handayani, K. Kusrini, and H. Al Fatta, "Perbandingan Fitur Ekstraksi Untuk Klasifikasi Emosi Pada Sosial Media," *J. Ilm. SINUS*, vol. 18, no. 2, p. 21, 2020, doi: 10.30646/sinus.v18i2.457.

[2] A. Musyayyidin and S. Adinugroho, "Analisis Emosional Pelajar terhadap Pembelajaran Daring Dengan Menggunakan Latent Semantic Indexing (LSI) dan N-Gram," vol. 5, no. 7, pp. 3013–3017, 2021.

[3] S. G. Tesfagergish, J. Kapočiūtė-Dzikienė, and R. Damaševičius, "Zero-Shot Emotion Detection for Semi-Supervised Sentiment Analysis Using Sentence Transformers and Ensemble Learning," *Appl. Sci.*, vol. 12, no. 17, 2022, doi: 10.3390/app12178662.

[4] D. H. F. Alan Tusa Bagus W, "Klasifikasi Emosi Pada Teks Dengan Menggunakan Metode Deep Learning," *J. Ilm. Indones. p-ISSN 2541-0849;Cirebon*, vol. 6, no. 1, 2021.

[5] R. Klinger, O. De Clercq, S. M. Mohammad, and A. Balahur, "IEST: WASSA-2018 Implicit Emotions Shared Task," *arXiv (Cornell Univ.*, 2018, doi: 10.48550/arxiv.1809.01083.

[6] A. Nurkasanah and M. Hayaty, "Feature Extraction using Lexicon on the Emotion Recognition Dataset of Indonesian Text," *Ultim. J. Tek. Inform.*, vol. 14, no. 1, pp. 20–27, 2022, doi: 10.31937/ti.v14i1.2540.

[7] A. N. Rohman, E. Utami, and S. Raharjo, "Deteksi Kondisi Emosi pada Media Sosial Menggunakan Pendekatan Leksikon dan Natural Language Processing," *Eksplora Inform.*, vol. 9, no. 1, pp. 70–76, 2019, doi: 10.30864/eksplora.v9i1.277.

[8] I. M. Abduh and H. Cangara, "Kritik Sosial Kebijakan Pemerintah dalam Platform Media Sosial dengan Pendekatan Komunikasi Hyperpersonal," *J. Nomosleca*, vol. 8, no. 1, pp. 91–100, 2022, doi: 10.26905/nomosleca.v8i1.7085.

[9] A. A. Efat, A. Atiq, A. S. Abeed, A. Momin, and M. G. R. Alam, "EMPOLITICON: NLP and ML Based Approach for Context and Emotion Classification of Political Speeches From Transcripts," *IEEE Access*, vol. 11, no. May, pp. 54808–54821, 2023, doi: 10.1109/ACCESS.2023.3282162.

[10] R. Olusegun, T. Oladunni, H. Audu, Y. A. O. Houkpati, and S. Bengesi, "Text Mining and Emotion Classification on Monkeypox Twitter Dataset: A Deep Learning-Natural Language Processing (NLP) Approach," *IEEE Access*, vol. 11, no. March, pp. 49882–49894, 2023, doi: 10.1109/ACCESS.2023.3277868.

[11] P. W. A. Wibawa and C. Pramartha, "Systematic

Literature Review: Machine Learning Methods in Emotion Classification in Textual Data," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 12, no. 3, pp. 425–433, 2023, doi: 10.32736/sisfokom.v12i3.1787.

[12] Z. Wu, "Research on Automatic Classification Method of Ethnic Music Emotion Based on Machine Learning," *J. Math.*, vol. 2022, 2022, doi: 10.1155/2022/7554404.

[13] F. Febriningsih, "Umpatan dalam Bahasa Melayu Ternate di Media Sosial," *Gramatika*, vol. VIII, no. 2, pp. 184–193, 2020.

[14] P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," *Soc. Netw. Anal. Min.*, vol. 11, no. 1, pp. 1–19, 2021, doi: 10.1007/s13278-021-00776-6.

[15] M. P. Solanki, "A Study on Emotion Detection & Classification from Text using Machine Learning," *J. Image Process. Intell. Remote Sens.*, no. 23, pp. 24–30, 2022, doi: 10.55529/jipirs.23.24.30.

[16] K. B. Rashmi, H. S. Guruprasad, and B. R. Shambhavi, "Sentiment Classification on Bilingual Code-Mixed Texts for Dravidian Languages using Machine Learning Methods," *CEUR Workshop Proc.*, vol. 3159, no. April 2023, pp. 899–907, 2021.

[17] S. Al-Saqqa, H. Abdel-Nabi, and A. Awajan, "A Survey of Textual Emotion Detection," *IEEE Xplore*. pp. 136–142, 2018. doi: 10.1109/CSIT.2018.8486405.

[18] M. A. H. Wadud, M. F. Mridha, and M. M. Rahman, "Word Embedding Methods for Word Representation in Deep Learning for Natural Language Processing," *Iraqi J. Sci.*, vol. 63, no. 3, pp. 1349–1361, 2022, doi: 10.24996/ijs.2022.63.3.37.

[19] S. Akuma, T. Lubem, and I. T. Adom, "Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets," *Int. J. Inf. Technol.*, vol. 14, no. 7, pp. 3629–3635, 2022, doi: 10.1007/s41870-022-01096-4.

[20] A. Majeed, H. Mujtaba, and M. O. Beg, "Emotion detection in roman urdu text using machine learning," *Proc. - 2020 35th IEEE/ACM Int. Conf. Autom. Softw. Eng. Work. ASEW 2020*, pp. 125–130, 2020, doi: 10.1145/3417113.3423375.

[21] S. Diantika, "Penerapan Teknik Random Oversampling Untuk Mengatasi Imbalance Class Dalam Klasifikasi Website Phishing Menggunakan Algoritma Lightgbm," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 19–25, 2023, doi: 10.36040/jati.v7i1.6006.

[22] H. D. Abubakar and M. Umar, "Sentiment Classification: Review of Text Vectorization Methods: Bag of Words, Tf-Idf, Word2vec and Doc2vec," *SLU J. Sci. Technol.*, vol. 4, no. 1&2, pp. 27–33, 2022, doi: 10.56471/slujst.v4i.266.

[23] D. I. Af'idah, D. Dairoh, S. F. Handayani, and R. W. Pratiwi, "Pengaruh Parameter Word2Vec terhadap Performa Deep Learning pada Klasifikasi Sentimen," *J. Inform. J. Pengemb. IT*, vol. 6, no. 3, pp. 156–161, 2021, doi: 10.30591/jpit.v6i3.3016.

[24] S. Guo, S. Wang, M. Wei, R. Chen, C. Guo, and H. Li, "Combining Imbalance Learning Strategy and Multiclassifier Estimator for Bug Report Classification," *Math. Probl. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/5712461.

[25] Zein Hanni Pradana, Hanin Nafi'ah, and Raditya Artha Rochmanto, "in Chatbot-based Information Service using RASA Open-SourceFrameworkin Prambanan Temple Tourism Object," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 4, pp. 656–662, 2022, doi: 10.29207/resti.v6i4.3913.

[26] A. Kadhim, "An Evaluation of Preprocessing Techniques for Text Classification," *https://www.researchgate.net/publication/329339664_An_Evaluation_of_Preprocessing_Techniques_for_Text_Classification*. International Journal of Computer Science and Information Security, 2018. [Online]. Available: https://sites.google.com/site/ijcsis/