Indonesian Twitter Emotion Recognition Model using Feature Engineering

Rhio Sutoyo[1], Harco Leslie Hendric Spits Warnars[2], Sani Muhamad Isa[3], Widodo Budiharto[4]
Computer Science Department-BINUS Graduate Program-Doctor of Computer Science,
Bina Nusantara University, Jakarta, Indonesia 11480[1,2]
Computer Science Department-BINUS Graduate Program-Master of Computer Science,
Bina Nusantara University, Jakarta, Indonesia 11480[3]
Computer Science Department-School of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480[4]

***Abstract*—Twitter is a social media platform that has a large amount of unstructured natural language text. The content of Twitter can be utilized to capture human behavior via emphasized emotions located in tweets. In their tweets, people commonly express emotions to show their feelings. Hence, it is crucial to recognize the text's underlined emotions to understand the message's meaning. Feature engineering is the process of improving raw data into often overlooked features. This research explores feature engineering techniques to find the best features for building an emotion recognition model on the Indonesian Twitter dataset. Two different text data representations were used, namely, TF-IDF and word embedding. This research proposed 12 feature engineering configurations in TF-IDF by combining data stemming, data augmentation, and machine learning classifiers. Moreover, this research proposed 27 feature engineering configurations in word embedding by combining three-word embedding models, three pooling techniques, and three machine-learning classifiers. In total, there are 39 feature engineering combinations. The configuration with the best F_1 score is TF-IDF with logistic regression, stemmed dataset, and augmented dataset. The model achieved 65.27% accuracy and 66.09% F_1 score. The detailed characteristics from the top seven models in TF-IDF also follow the same feature engineering configuration. Lastly, this work improves performance from the previous research by 1.44% and 2.01% on the word2vec and fastText approaches, respectively.**

***Keywords*—Text classification; feature engineering; emotion recognition; Indonesian tweet; natural language processing**

I. Introduction

Twitter is a social media platform that provides services to share ideas and opinions. The popularity of Twitter leads to millions of users sending data in the form of tweets, i.e. a short message from Twitter users [1]. As a result, Twitter has a large amount of unstructured natural language text. Researchers have used the content of Twitter to predict economic trends, e.g., financial market prediction [2]. Furthermore, Twitter data can also be utilized to capture human behavior via emphasized emotions located in tweets.

Based on Shaver's theory, emotions can be categorized into five basic classes, i.e., anger, fear, happiness, love, and sadness [3]. On Twitter, people commonly express emotions to show their feelings toward something, e.g., political party, sexual abuse, or simply a bitter experience in their life. Thus, it is crucial to recognize the text's underlined emotion to understand the message's meaning [4]. The underlined emotions can be identified directly via emotion words, e.g., *dukacita* (grief) for sadness, and *kesal* (annoyed) for anger. However, Twitter users can also implicitly display emotions in their tweets, making them hard to identify.

The ability to recognize emotions automatically is essential for various applications. In politics, emotion recognition can predict the polarity of the sentiment in the Presidential election based on social media data [5]. Emotion recognition can also be utilized for games with emotion-based dynamic difficulty adjustment [6]. Furthermore, emotion recognition can also be utilized to build a recommendation system for culinary and food [7]. Lastly, the emotion model can be used to build an emotionally aware chatbot capable of recognizing and interpreting human emotions [8].

There are several challenges to building an automatic emotion recognition model. First, the natural language text in Twitter data is unstructured and uncontrolled, e.g., the length of tweets might be too short or too long, the texts contain typos, and the texts contain misused terms. Second, Twitter datasets for emotion recognition tasks are primarily available for the English language [9], [10]. Datasets for Indonesian emotion recognition from previous studies are not available publicly [11], [12]. Fortunately, Saputri et al. share their Indonesian Twitter dataset for emotion classification task [13].

Inspired by the work of [13], the authors explore feature engineering to build an emotion recognition model on the Indonesian Twitter dataset. This work extends the work of [13] by further exploring feature engineering techniques to find the best features for identifying emotion in Indonesian Tweet. It combines various data preprocessing techniques, word embedding models, different pooling techniques, and machine learning classifiers. The limitation of this work is it does not include an experiment using a combination of features. Specifically, this work focuses on basic features, namely, Bag-of-Words (BOW), Word2Vec (WV), and FastText (FT).

In total, 39 feature engineering configurations are proposed and compared. This work compares the best feature engineering configurations based on the F_1 score metric for evaluation. The best F_1 score achieved was 66.09% from the TF-IDF approach with logistic regression, stemmed dataset, and augmented dataset. The experiment results have shown that the TF-IDF text representation is better than word embedding. On average, the stemming process increases the accuracy performance by 0.11%. Nevertheless, it decreases the F_1 score by 0.06%. Moreover, the dataset augmentation reduces both

accuracy and F_1 score by 0.45% and 0.67%, respectively. These results contribute to comprehensively exploring feature engineering techniques to identify the best features for building emotion recognition models on the Indonesian Twitter dataset [13]. It also has potential practical implications for developing an emotionally aware chatbot capable of recognizing human emotions.

The structure of this paper is as follows: related works are presented in Section II. The system architecture of this work is described in Section III. The results and discussions are presented in Section IV and Section V, respectively. Lastly, the conclusion and future work of this work are discussed in Section VI.

II. Related Works

A. Emotion Recognition

Emotion recognition is an attempt to recognize and classify people's emotions. The basic pipeline of emotion recognition is data preprocessing, data representation, training model, and model evaluation. There are two superior-level categories of emotions: positive and negative [3]. Two major basic level categories in the positive category were love and happiness. The three major basic level categories in the negative category were anger, fear, and sadness. Emotion recognition tasks can be performed in various modalities, such as textual [13], audio [14], speech [15], and brain activity [16]. Furthermore, it can also be performed in a multimodal dataset [4].

B. Dataset of Emotion Recognition

Several datasets can be utilized to train the emotional recognition model. The ISEAR [17], the Tales [18], and the AffectiveText [19] are known datasets that are available in the English language. The ISEAR consists of 7,665 sentences labeled with a specific emotion, i.e., joy, fear, anger, sadness, disgust, shame, and guilt [17]. The Tales consists of 15,302 sentences from 176 stories by three different authors [18]. It utilizes Ekman's six basic emotions theory [20], merging anger and disgust. The AffectiveText consists of 1,250 instances from news headlines and has six basic emotions models of Ekman complemented by its valence [19]. Furthermore, the emotions recognition dataset is also available in the Indonesian language, i.e. the Indonesian Twitter Emotion Dataset [13]. It consists of 4,403 Indonesian tweets with general topics. The emotions model is labeled using Shaver's five basic emotions: love, anger, sadness, and fear [3]. The summary of the emotion recognition dataset is listed in Table I.

The study of [13] has performed feature engineering using various text features: lexicon-based, Bag-of-Words, word embeddings, orthography, and Part-of-Speech (POS) tags. In the experiment, [13] has represented word embeddings with several dimensional variations. However, the research has not focused on exploring feature engineering with various word embedding pooling techniques. This paper addresses that issue by exploring various feature engineering configurations, including using various word embedding pooling techniques, to build an Indonesian emotion recognition model.

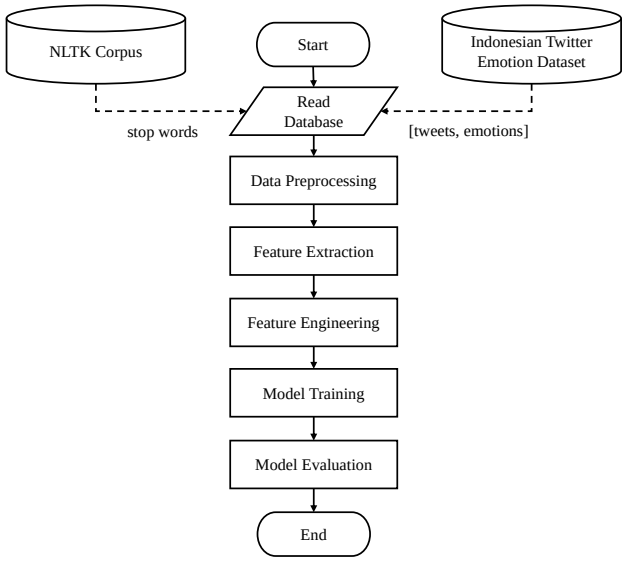

Fig. 1. The system architecture of Indonesian twitter emotions recognition using feature engineering.

III. System Architecture

This work uses feature engineering to present a system architecture for Indonesian Twitter emotion recognition (see Fig. 1). There are two sources of the dataset: NLTK Corpus[1] for a list of Indonesian stop words and Indonesian Twitter Emotions Dataset [13] for model training.

The arrow in the system architecture shows the direction of the process. Furthermore, the dashed arrow shows the data flow. The details of each step in building our model are explained in the following sections:

A. NLTK Corpus

In data preprocessing, the system removes stop words from text inputs. Stops words are low-value words that generally do not contain helpful sentence meanings. The library of NLTK Corpus contains approximately 750 Indonesian stop words. Examples of stop words are *adalah* (is), *agak* (somewhat), and *ke* (to).

B. Indonesian Twitter Emotion Dataset

This work utilizes the Indonesian Twitter emotion dataset from [13] for model training. Saputri et al. collected 4,403 Indonesian tweets using Twitter Streaming API for two weeks. The Twitter metadata, namely, username, hyperlink, and phone number in the sentences, are converted into special tags (i.e., [USERNAME], [URL], and [SENSITIVE-NO]). Then, they annotated the collected dataset with Shaver's emotions model [3]. The emotion distribution of the dataset is shown in Fig. 2.

[1]https://www.nltk.org/api/nltk.corpus.html

TABLE I. PUBLICLY AVAILABLE EMOTION RECOGNITION DATASET

No	Dataset Name	Size	Granularity	Language
1	ISEAR [17]	7,665	descriptions	English
2	Tales [18]	15,302	sentences	English
3	AffectiveText [19]	1,250	headlines	English
4	Indonesian Twitter Emotion Dataset [13]	4,403	tweets	Indonesian

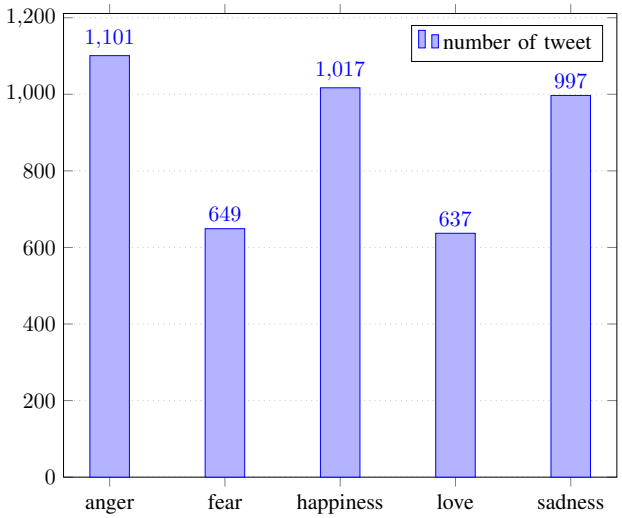

Fig. 2. The class distribution of Indonesian twitter emotion dataset.

C. Read Database

This section explains the process of reading and utilizing data from the data sources. The NLTK corpus was utilized to get Indonesian stop words. Then, the list of stop words was used in the data preprocessing step. The Indonesian Twitter emotion dataset is stored in Pandas DataFrame. Then, the data were preprocessed and converted to features for the model training.

D. Data Preprocessing

Data is the training material for the emotions recognition model. The quality of the model depends on the readiness of the data. Hence, the data preprocessing prepares data for the model training by performing several improvement steps. The case-folding converts all sentences to lowercase to avoid counting the same words as different information. Irrelevant information (i.e. username, URL, sensitive number) is removed because it is unrelated to the emotion recognition task. Then, the inputs were standardized using regular expressions to keep only the alphabet a – z. The stemming algorithm converts words to their roots. For this task, the Sastrawi python library was applied, utilizing the algorithm of Nazief and Adriani [21]. The stemming process was executed in approximately 25 minutes on 17,903 tokens. Lastly, stop words were removed to focus on important words. As a result, the data preprocessing reduces the tokens to 13,521. The algorithm illustration for the data preprocessing is shown in Algorithm 1.

E. Feature Extraction

This work utilizes two types of word representation for text analysis: TF-IDF and word embedding. The TF-IDF represents a document as a vector. On the other hand, word embedding represents a word as a vector. Furthermore, the length of array word embedding has a fixed array length, i.e., 300 dimensions. In the TF-IDF, the array's length depends on the size of the bag-of-words (BoW).

Algorithm 1 Algorithm for data preprocessing

```
corpus = [ ]
i ← 0
N ← len(tweets)
while i < N do
    tweet ← tweets[i]                                  ▷ take a single tweet
    tweet = tweet.lower()                              ▷ case folding
    tweet = tweet.removeIrrelevantInformation()
    tweet = re.sub('[^a-z]+',' ',tweet)                ▷ standardization
    tweet = stemmer.stem(tweet)                        ▷ stemming
    tweet = tweet.removeStopWords()
    corpus.append(tweet)                               ▷ combine tweet
end while
```

Three models of word embedding are utilized in this work. First, word2vec consists of 129,390 vocabulary sizes with 400 sizes of vector [13]. Second, fastText consists of 69,465 sizes of vocabulary with 100 sizes of vector [13]. Lastly, the neural network language modeling (NNLM) architecture from Google has 128 vector sizes[2]. It is trained on the Indonesian Google News 3B corpus.

In the word embedding approach, each word vector from a tweet is pooled into one vector. This work utilizes three techniques: mean pooling, sum pooling, and min-max pooling. The sum pooling sums up all vectors into the pooled vector. The mean pooling sums up all vectors and then finds their average value as the pooled vector. Lastly, the min-max pooling combines the minimum vector with the maximum vector. The result of min-max pooling is double the size of a vector. The word embedding vectorization and the pooling technique are presented in Fig. 3.

F. Feature Engineering

This research performs and combines several text-processing techniques to extract features from texts for model training. Two different text data representations were used, namely, TF-IDF and word embedding. The dataset was split into a train set (80%) and a test set (20%). This work uses a stratified train-test method to ensure both sets have a proportioned class distribution. Furthermore, a random seed '88' was used to ensure reproducible results.

SMOTE was used to perform data augmentation, i.e., over-sampling the data for model training [22]. The synthetic data were created at the vector level. The data was sampled by using a maximum sampling strategy. Moreover, a fixed random seed, i.e., '88', was used consistently to get the same sampling

[2]https://tfhub.dev/google/tf2-preview/nnlm-id-dim128/1

Fig. 3. The illustration of word embedding and pooling process.

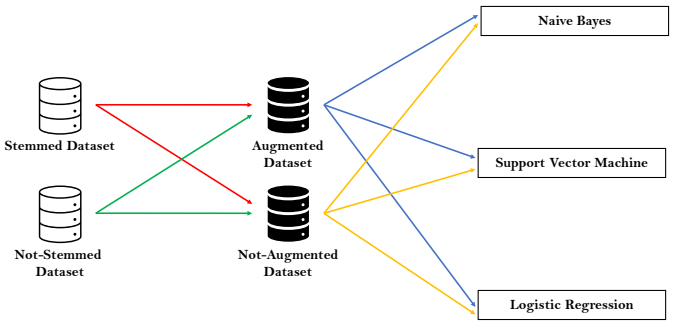

Fig. 4. Feature engineering configurations in TF-IDF.

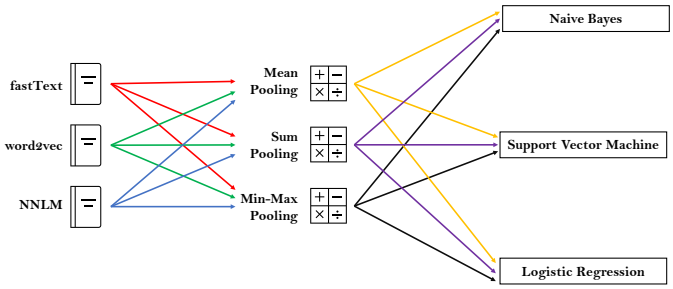

Fig. 5. Feature engineering configurations in word embedding.

result. Different machine learning classifiers were utilized for training the emotion recognition model. Based on the preliminary research, the recommended classifiers are Naive Bayes (NB), Logistic Regression (LR), and Support Vector Machine (SVM).

This research proposes twelve feature engineering configurations in TF-IDF by mixing data stemming, data augmentation, and machine learning classifiers. Fig. 4 shows the feature engineering configuration illustration for TF-IDF. The red line symbolizes the data flow of the stemmed dataset, and the green line symbolizes the data flow of the not-stemmed dataset. Then, those data were passed to the data augmentation process. The blue lines symbolize the data flow of the augmented dataset, and the yellow line symbolizes the data flow of the not-augmented dataset. Finally, those data were passed to three different classifiers.

This research proposes 27 feature engineering configurations in word embedding by mixing three-word embedding models, three pooling techniques, and three machine-learning classifiers. Fig. 5 shows the feature engineering configuration illustration for word embedding. The red, green, and blue lines symbolize feature extraction using fastText, word2vec, and NNLM, respectively. Then, the word embedding results were combined into sentence embedding with three different pooling techniques. The yellow, purple, and black lines symbolize the pooling technique using mean pooling, sum pooling, and min-max pooling, respectively. Finally, the data were passed to three different classifiers.

This work creates a naming scheme to differentiate the configurations. For example, the configuration of TF-IDF with SVM, not-stemmed database, and augmented is "TFIDF_SVM_notstem_aug". Moreover, the configuration of word embedding with Naive Bayes, fastText, and sum pooling is "WE_NB_ft_sum". The detail of the naming scheme is as follows:

- **TF-IDF**: TFIDF_name of classifier technique_status of dataset stemming_status of dataset augmentation.
- **Word Embedding**: WE_classifier technique_word embedding model_pooling technique.

G. Model Training

The emotion recognition model was trained using Google Colab[3]. The model training uses '88' as the random seed. Furthermore, the model training was performed using Python v3.7.13, NLTK v3.7, scikit-learn v1.0.2, and various text processing libraries (e.g., re for regular expression operations).

H. Model Evaluation

The emotion recognition model is evaluated using accuracy and macro F_1 score. The accuracy score is calculated by dividing the number of correct predictions by the number of total data. The equation of accuracy metric is presented in Equation (1).

[3]http://colab.research.google.com/

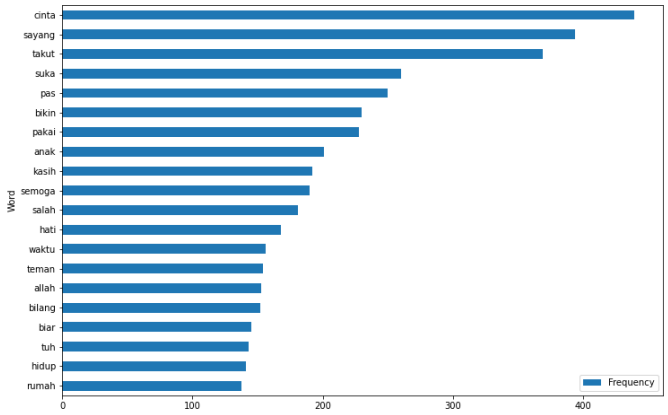

Fig. 6. Word frequency analysis of top 20 common words (excluding stop words).

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

In machine learning, TP means true positive, TN means true negative, FP means false positive, and FN means false negative.

$$F_1 = \frac{2 * TP}{2 * TP + FP + FN} \quad (2)$$

The F_1 metric combines precision and recall to calculate the harmonic mean of the two metrics. The equation of F_1 metric is presented in Equation (2).

IV. Results

This section discusses the result of exploring the Indonesian Twitter emotion dataset. Furthermore, the result of the experiment is divided into two subsections, namely TF-IDF and Word Embedding. The full results of the experiment can be viewed on the GitHub page[4].

A. Dataset Exploration

Fig. 6 shows the word frequency analysis of the top 20 common words in the Indonesian Twitter emotion dataset. The result has excluded stop words that are valueless for model training. It shows the words "*cinta*" (love), "*sayang*" (darling), "*takut*" (afraid), and "*suka*" (like) are the most commonly occurring word frequencies.

Moreover, the frequent use of words is also shown as a word cloud. The diagram below includes all words from the dataset. It can be seen in Fig. 7 that the most common words are dominated by stop words, e.g., "*saya* (I)," "*yang* (which)," and "*kamu* (you)."

B. TF-IDF

This work explored 12 feature engineering configurations of TF-IDF by combining data stemming, data augmentation, and machine learning classifiers. The F_1 score results for every configuration with TF-IDF are presented in Fig. 8.

Fig. 7. Word cloud plot of Indonesian twitter emotion dataset (including all words).

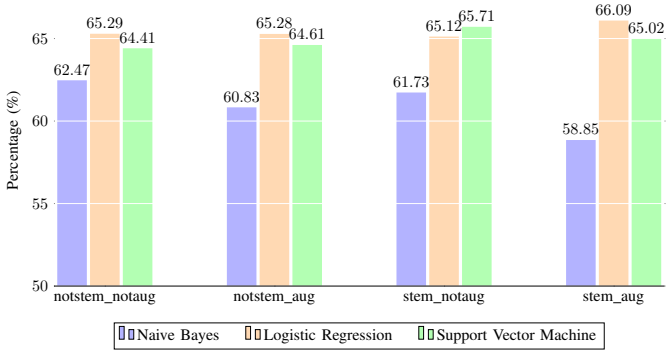

Fig. 8. The F_1 score results with TF-IDF configurations

(notstem: not-stemmed, stem: stemmed, notaug: not-augmented, aug: augmented)

The best configuration in the TF-IDF approach is "TFIDF_LR_stem_aug," i.e., the combination of logistic regression (LR), stemmed dataset (stem), and augmented dataset (aug). The model achieved 65.27% accuracy and 66.09% F_1 score. Furthermore, the worst configuration in the TF-IDF approach is from "TFIDF_NB_stem_aug" with 59.59% accuracy and 58.85% F_1 score. The experimental results show that different machine learning classifiers greatly impact the same preprocessed dataset, i.e., stemmed dataset (stem) and augmented dataset (aug). Moreover, the performance of the Naive Bayes classifier is inferior to the others.

From the machine learning classifier perspective, the logistic regression provides the best performance across all configurations, i.e., 64.42% average accuracy and 65.45% average F_1 score. From the data preprocessing perspective, using the stemming process increases the overall accuracy by 0.11% compared to not using it. However, it reduces the overall F_1 score by 0.06%. Finally, the augmentation process reduces average accuracy and average F_1 score by 0.45% and 0.67%, respectively.

Fig. 9 shows the confusion matrix result of the TF-IDF model, i.e., "TFIDF_LR_stem_aug". The most accurate prediction is anger (18.39%), and the least accurate prediction

[4]https://github.com/rhiosutoyo/emotion-recognition-model

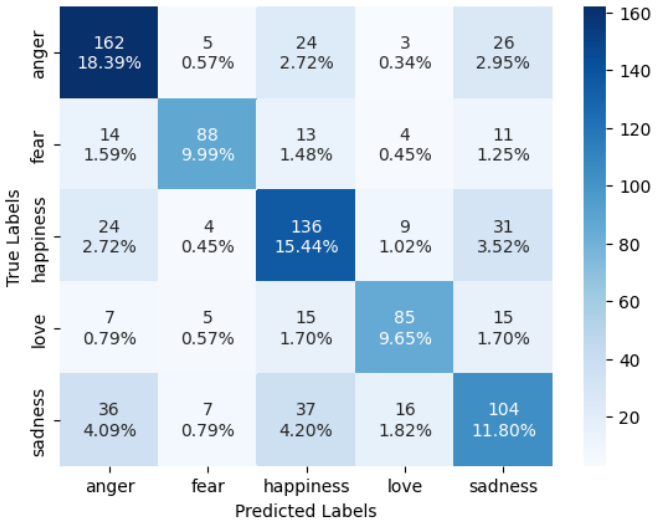

Fig. 9. The confusion matrix from the best TF-IDF model (TFIDF_LR_stem_aug).

is love (9.65%). The result is unsurprising because anger has the highest label than the other emotions. Furthermore, love has the lowest label than the other emotions. The confusion matrix result shows that anger tends to be misinterpreted as sadness. Moreover, fear tends to be misinterpreted as anger. Happiness tends to be misinterpreted as sadness. Love tends to be misinterpreted as happiness or sadness. Lastly, sadness tends to be misinterpreted as happiness.

Based on the experiment results, implementing data augmentation to increase the training data does not yield a positive outcome. In theory, the augmentation process is supposed to increase the model performance. However, the performance of using augmentation techniques is lower than that of not using them. This work argues that the original dataset has performed well because the Indonesian Twitter dataset is quite balanced. Thus, the augmented dataset is having trouble outperforming the performance of the not-augmented dataset.

C. Word Embedding

This work explored 27 feature engineering configurations of word embedding (WE) by combining three-word embedding models, three pooling techniques, and three machine-learning classifiers. The F_1 score results for every configuration with word embedding are presented in Fig. 10.

The best configuration in the word embedding approach is from the combination of support vector machine (SVM), fastText (ft), and sum pooling (sum), or "WE_SVM_ft_sum." The model achieved 65.27% accuracy and 64.50% F_1 score. Furthermore, the worst configuration in the word embedding approach is from "WE_NB_w2v_sum" with 38.59% accuracy and 36.86% F_1 score.

From the machine learning classifier perspective, the SVM performs best across all configurations, i.e., 58.17% average accuracy and 57.89% average F_1 score. From the word embedding model's perspective, the fastText model provides the best performance across all configurations, i.e., 56.26% average

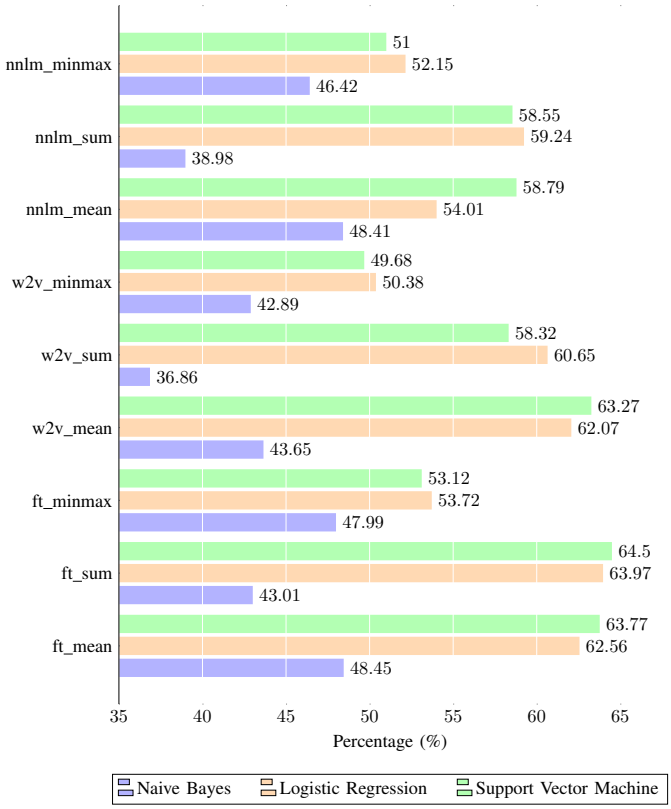

Fig. 10. The F_1 score results with word embedding configurations.

(nnlm: neural network language model, w2v: word2vec, ft: fastText)

accuracy and 55.68% average F_1 score. The fastText's average F_1 score is 3.52% higher than Google's NNLM and 3.7% higher than the word2vec model. The word2vec incapability to handle out-of-vocabulary tokens might be the reason for the poor performance. Lastly, the pooling technique with the best average performance across all configurations is mean pooling, i.e., 57.28% average accuracy and 56.83% average F_1 score. The mean pooling's average F_1 score is 7.12% higher than the min-max pool and 2.49% higher than the sum pool. The min-max pooling technique that discards features from the input might be the reason for the poor performance.

Fig. 11 shows the confusion matrix result of the word embedding model, i.e., "WE_SVM_ft_sum". The most accurate prediction is anger (21.11%), and the least accurate is sadness (8.29%). Emotion with the highest result from word embedding is the same as the TF-IDF approach,i.e., anger. Emotion with the lowest result from word embedding differs from the TF-IDF approach, i.e., love. The confusion matrix result shows that anger tends to be misinterpreted as happiness. Furthermore, fear tends to be misinterpreted as anger. Happiness tends to be misinterpreted as anger. Love tends to be misinterpreted as happiness. This is normal because both emotions share similar characteristics. Lastly, sadness tends to be misinterpreted as anger.

Unlike the TF-IDF, the confusion matrix result of word embedding shows that the prediction results do not align with the sum of the label quantities in the dataset. Based on the sum

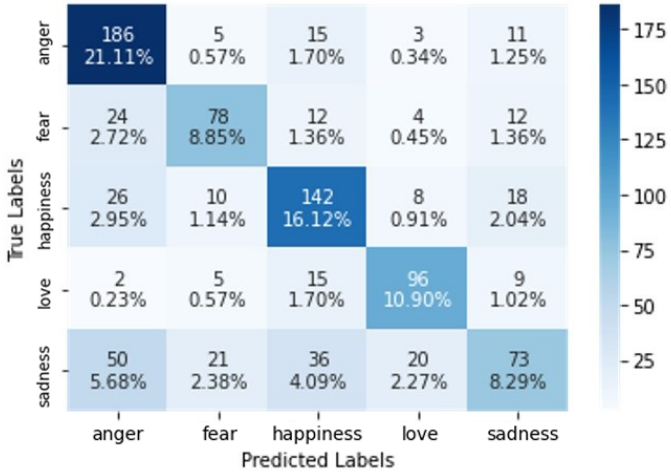

Fig. 11. The confusion matrix from the best word embedding model (WE_SVM_ft_sum).

TABLE II. THE CONFIGURATION DETAIL OF THE TOP SEVEN MODELS IN TF-IDF

Configuration Component	Measurement	Frequency	%
Data Stemming	Stemmed	4	57%
	Not-Stemmed	3	43%
	Total	7	100%
Data Augmentation	Augmented	4	57%
	Not-Augmented	3	43%
	Total	7	100%
Machine Learning Classifier	Logistic Regression	4	57%
	Support Vector Machine	3	43%
	Naive Bayes	0	0%
	Total	7	100%

of the label quantities in the dataset, the order of emotions is anger, happiness, sadness, fear, and love (see Fig. 2). Based on the confusion matrix performance, the emotions are anger, happiness, love, fear, and sadness (see Fig. 11).

V. DISCUSSIONS

A. Configuration Detail of the Top Seven Models

This work proposed 39 feature engineering configurations, i.e., 12 with TF-IDF and 27 with word embedding. This section shows the configuration detail of the top seven models from both word representation techniques.

The configuration detail of the top seven models in TF-IDF is shown in Table II. The experiment shows that the top-performance model has the following component configurations: logistic regression, stemmed dataset, and augmented dataset. The result matches the top model on TF-IDF, which is "TFIDF_LR_stem_aug."

Moreover, the configuration detail of the top seven models in word embedding is shown in Table III. The experiment shows that the top-performance model has the following component configurations: logistic regression, fastText, and mean pooling. The result does not match the top word embedding model, "WE_SVM_ft_sum."

The top seven model configurations of TF-IDF and word embedding produce an F_1 score of more than 60%. On

TABLE III. THE CONFIGURATION DETAIL OF THE TOP SEVEN MODELS IN WORD EMBEDDING

Configuration Component	Measurement	Frequency	%
Word Embedding Model	fastText	4	57%
	word2vec	3	43%
	NNLM	0	0%
	Total	7	100%
Pooling Technique	Mean Pooling	4	57%
	Sum Pooling	3	43%
	Min-Max Pooling	0	0%
	Total	7	100%
Machine Learning Classifier	Logistic Regression	4	57%
	Support Vector Machine	3	43%
	Naive Bayes	0	0%
	Total	7	100%

TABLE IV. AVERAGE F_1 SCORE OF TF-IDF AND WORD EMBEDDING FROM THE TOP SEVEN MODEL CONFIGURATIONS

Word Representation	F_1 Score
TF-IDF	65.30%
Word Embedding	62.97%

average, the top seven models of TF-IDF perform better than word embedding. The results are shown in Table IV.

B. TF-IDF vs. Word Embedding

Based on the experiment, the F_1 score from the best model of the word embedding technique (64.5%) is lower than the TF-IDF technique (66.09%). Hence, the best feature engineering configuration for the emotion recognition model is the TF-IDF approach from the combination of logistic regression (LR), stemmed dataset (stem), and augmented dataset (aug).

In theory, the word embedding technique should be able to produce a higher performance result because it has dense information packed in fixed-size arrays. Nevertheless, other research also shows that the performance of word embedding is lower than TF-IDF [23], [24]. In their research [23], Piskorski and Jacquet argue that features from word embedding might be great for deep learning but not for machine learning. Furthermore, specific features make the dataset biased in favor of traditional machine-learning approaches. These features appear exclusively for some categories (e.g., unique keywords). Thus, the classical machine learning algorithms can perform the classification with high precision because of the feature vector built using the TF-IDF technique.

C. Performance Comparison

The previous work from [13] does not provide codes and test splits. Hence, this study repartitioned the Indonesian Twitter emotion dataset by using several random seed values to perform the experiments. Ultimately, the random seed "88" was chosen because it achieved the best result.

The experiment resulted in a slightly better performance than the previous study [13] from the perspective of basic features, i.e., Word2Vec (WV) and FastText (FT). The Word2Vec (WV) F_1 score is increased by 1.44% by using mean pooling and SVM. Moreover, the FastText (FT) F_1 score is increased by 2.01% by using sum pooling and SVM.

In general, the quality of features can be increased by utilizing different pooling methods. Moreover, SVM produces better results than logistic regression.

VI. Conclusions and Future Work

This research explored feature engineering to build an emotion recognition model on the Indonesian Twitter dataset. Two different text data representations were used, namely TF-IDF and word embedding. This research proposes 12 feature engineering configurations in TF-IDF by mixing data stemming, data augmentation, and machine learning classifiers. Furthermore, this research proposes 27 feature engineering configurations in word embedding by mixing three-word embedding models, three pooling techniques, and three machine-learning classifiers. Moreover, this research analyzed the top seven models of both data representation techniques to find the recommended configuration component. Finally, performance comparisons were conducted to evaluate the models further.

The best performance configuration of TF-IDF is achieved by "TFIDF_LR_stem_aug," i.e., logistic regression (LR), stemmed dataset (stem), and augmented dataset (aug). The model achieved 65.27% accuracy and 66.09% F_1 score. The best performance configuration of word embedding is achieved by "WE_SVM_ft_sum," i.e., support vector machine (SVM), fastText (ft), and sum pooling (sum). The model achieved 65.27% accuracy and 64.50% F_1 score. The detailed characteristics from the top seven models show the recommended component configurations in TF-IDF: logistic regression, stemmed dataset, and augmented dataset. The recommended component configurations in word embedding are logistic regression, fastText, and mean pooling.

Furthermore, the experiment shows a slightly better performance than the previous study from the perspective of a single basic feature, i.e., word embedding. This work improved the word2vec F_1 score by 1.44% by using mean pooling and SVM. Moreover, the fastText F_1 score is increased by 2.01% by using sum pooling and SVM. Based on the results, the quality of text features in word embedding can be enhanced by utilizing different pooling methods. The recommended configuration element in word2vec is mean pooling; in fastText, it is sum pooling.

Lastly, the experiment shows that word embedding performs lower than TF-IDF. Thus, further exploration of utilizing word embedding in deep learning with a more significant number of examples can become the focus of future work.

References

[1] A. S. Girsang, S. M. Isa, and I. Harvy, "Recommendation System Journalist For Getting Top News Based On Twitter Data," *J. Phys. Conf. Ser.*, vol. 1807, no. 1, p. 012006, Apr. 2021.

[2] X. Guo and J. Li, "A novel twitter sentiment analysis model with baseline correlation for financial market prediction with improved efficiency," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE, 2019, pp. 472–477.

[3] P. R. Shaver, U. Murdaya, and R. C. Fraley, "Structure of the indonesian emotion lexicon," *Asian journal of social psychology*, vol. 4, no. 3, pp. 201–224, 2001.

[4] G. Mohammadi and P. Vuilleumier, "A multi-componential approach to emotion recognition and the effect of personality," *IEEE Transactions on Affective Computing*, 2020.

[5] W. Budiharto and M. Meiliana, "Prediction and analysis of indonesia presidential election from twitter using sentiment analysis," *Journal of Big data*, vol. 5, no. 1, pp. 1–10, 2018.

[6] A. Andrew, A. N. Tjokrosetio, and A. Chowanda, "Dynamic difficulty adjustment with facial expression recognition for improving player satisfaction in a survival horror game," *ICIC Express Letters*, vol. 14, no. 11, pp. 1097–1104, 2020.

[7] B. Siswanto, F. L. Gaol, B. Soewito, and H. L. H. S. Warnars, "Sentiment analysis of big cities on the island of java in indonesia from twitter data as a recommender system," in *2021 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*. IEEE, 2021, pp. 124–128.

[8] R. Sutoyo, H. L. H. S. Warnars, S. M. Isa, and W. Budiharto, "Emotionally aware chatbot for responding to indonesian product reviews," *International Journal of Innovative Computing, Information and Control*, vol. 19, no. 03, p. 861, 2023.

[9] F. R. Lapitan, R. T. Batista-Navarro, and E. Albacea, "Crowdsourcing-based annotation of emotions in filipino and english tweets," in *Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016)*, 2016, pp. 74–82.

[10] S. Mohammad and F. Bravo-Marquez, "WASSA-2017 shared task on emotion intensity," in *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, A. Balahur, S. M. Mohammad, and E. van der Goot, Eds. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 34–49. [Online]. Available: https://aclanthology.org/W17-5205

[11] N. A. S. Winarsih, C. Supriyanto *et al.*, "Evaluation of classification methods for indonesian text emotion detection," in *2016 International seminar on application for technology of information and communication (ISemantic)*. IEEE, 2016, pp. 130–133.

[12] K. S. Nugroho and F. A. Bachtiar, "Text-based emotion recognition in indonesian tweet using bert," in *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. IEEE, 2021, pp. 570–574.

[13] M. S. Saputri, R. Mahendra, and M. Adriani, "Emotion classification on indonesian twitter dataset," in *2018 International Conference on Asian Language Processing (IALP)*. IEEE, 2018, pp. 90–95.

[14] N. Rosli, N. Rajaee, and D. Bong, "Renica based music source separation for automatic music emotion classification," *INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING INFORMATION AND CONTROL*, vol. 14, no. 6, pp. 2325–2333, 2018.

[15] D. Wu, H. Zhao, X. Zhang, Z. Tao, and C. Huang, "Multi-scaled emotional recognition from speech using fuzzy model and markov random fields based configuration," *ICIC Express Lett.*, vol. 9, no. 6, pp. 1637–1642, Jan. 2015.

[16] H. Jung, M. Kwon, and H.-I. Cheng, "Analysis of EEG for violent movies," *ICIC Express Lett.*, vol. 10, no. 7, pp. 1523–1528, Jul. 2016.

[17] E. S. Dan-Glauser and K. R. Scherer, "The difficulties in emotion regulation scale (ders)," *Swiss Journal of Psychology*, 2012.

[18] C. O. Alm, D. Roth, and R. Sproat, "Emotions from text: machine learning for text-based emotion prediction," in *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, 2005, pp. 579–586.

[19] C. Strapparava and R. Mihalcea, "Semeval-2007 task 14: Affective text," in *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, 2007, pp. 70–74.

[20] D. A. Sauter, F. Eisner, P. Ekman, and S. K. Scott, "Cross-cultural recognition of basic emotions through nonverbal emotional vocalizations," *Proceedings of the National Academy of Sciences*, vol. 107, no. 6, pp. 2408–2412, 2010.

[21] M. Adriani, J. Asian, B. Nazief, S. M. Tahaghoghi, and H. E. Williams, "Stemming indonesian: A confix-stripping approach," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 6, no. 4, pp. 1–33, 2007.

[22] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[23] J. Piskorski and G. Jacquet, "Tf-idf character n-grams versus word embedding-based models for fine-grained event classification: a preliminary study," in *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*, 2020, pp. 26–34.

[24] A. W. Romadon, K. M. Lhaksmana, I. Kurniawan, and D. Richasdy, "Analyzing tf-idf and word embedding for implementing automation in job interview grading," in *2020 8th International Conference on Information and Communication Technology (ICoICT)*. IEEE, 2020, pp. 1–4.