Sentence-Level Granularity Oriented Sentiment Analysis of Social Media Using Long Short-Term Memory (LSTM) and IndoBERTweet Method

Nisa Maulia Azahra, Erwin Budi Setiawan
Telkom University, Jl. Terusan Buah Batu, Bandung 40257, Indonesia

ARTICLE INFO

Article history:

Received January 02, 2023
Revised February 06, 2023
Published February 09, 2023

Keywords:

Sentiment analysis sentence-level granularity oriented;
LSTM;
TF-IDF;
IndoBERTweet;
Word2Vec

ABSTRACT

The dissemination of information through social media has been rampant, especially on the Twitter platform. This information eventually invites various opinions from users as their points of view on a topic being discussed. These opinions can be collected and processed using sentiment analysis to assess public tendencies to obtain a fundamental source of decision-making. However, the procedure is not optimal enough due to its inability to recognize the word meaning of the opinion sentences. By using sentence-level granularity-oriented sentiment analysis, the system can explore the "sense of the word" in each sentence by giving it a granularity weight as the system's consideration in recognizing word meaning. To construct the procedure, this research utilizes LSTM as the classification model combined with TF-IDF and IndoBERTweet as feature extraction. Not only that, but this research also conducts the Word2Vec feature expansion method which was built using Twitter and IndoNews corpus to produce word similarity corpus and find effective word semantics. To be fully compliant with the granularity requirements, manual labeling, and system labeling were performed by considering weight granularity as a model performance comparison. This research succeeded in getting 88.97% accuracy for manual labeling data and 97.80% for system labeling data after combining these methods. The experimental results show that the granularity-oriented sentiment analysis model can outperform the conventional sentiment analysis system which can be seen based on the high performance of the resulting system.

Corresponding Author:

Erwin Budi Setiawan, Telkom University, Jl. Terusan Buah Batu, Bandung 40257, Indonesia
Email: erwinbudisetiawan@telkomuniversity.ac.id

1. INTRODUCTION

Twitter has evolved into a place to freely express perspectives on a topic through the comments feature [1]. Such comments are considered very important as they can be a source of information in assessing user responses to the topic under discussion [2], [3]. Responses are collected and classified into negative, neutral, and positive sentiment groups to serve as a source of decision-making [4]. However, the classification process is impossible to do manually for processing a large amount of data [5]. Therefore, a sentimsent analysis system is needed by utilizing deep learning architecture that can handle tasks with large amounts of data and high dimensions [6].

People often express their opinions based on the bias of a topic, making their opinions subjective [7]. This habit eventually carries over when commenting on social media, so the system needs to understand data with subjective sentences. By using sentence-level granularity sentiment analysis, the system can handle such cases by recognizing the "sense of the word" to reduce the possibility of system errors in classifying data inputs [8]. This method fulfills an additional need by weighting word values within a certain range as a representation of word meaning. One of these types is sentence-level which focuses more on subjective sentence identification [9].

Sentence-level granularity is divided into two problem classifications, the first one is known as Subjective Classification literature which distinguishes subjective sentences (opinions) from objective sentences (facts). Subjective sentences usually contain personal feelings and judgments that are different for each individual. Whereas objective sentences contain the same information and apply to all individuals. The second classification is called Sentiment Classification [2], [9]. This stage is done with a sentiment classification process that categorizes it into negative, neutral, and positive groups.

Sentiment analysis using LSTM model has been done in [10], [11]. Ling conducted granularity sentiment analysis using a hybrid C-LSTM model with TFIDF and ELMo. The dataset uses Chinese language crawled from Weibo microblog to be classified into positive and negative sentiments. This research gets 81.31% accuracy by using ELMo pre-trained model. Meanwhile, Bai [11] tried to classify the ABSC task dataset into positive, neutral, and negative sentiments using aspect-based sentiment analysis. The system was built using LSTM and GloVe models. The highest accuracy of using the LSTM model is at a value of 80.67% on the Restaurant16 dataset.

Hong [12] has also conducted research analyzing Twitter comments using BERT method to detect toxic sentences. BERT has been trained using English vocabulary through fine-tuning step with 12 encoder blocks and 768 hidden layers as its conFig.uration. This research managed to get an accuracy value of 98% for 14,000 data. Meanwhile, research using the IndoBERTweet method as a derivative of IndoBERT trained using Indonesian obtained from Twitter user comments obtained an average accuracy value of 86.1% [13].

Based on these studies, it can be concluded that the LSTM and IndoBERTweet models can be used to perform sentiment analysis on large amounts of data. As far as researchers know, there has been no research that conducts sentence-level granularity-oriented sentiment analysis in Indonesian. Thus, this study was conducted to determine the effectiveness of sentence-level granularity-oriented sentiment analysis systems in processing data containing opinion sentences compared to using conventional sentiment analysis procedures. In addition, this study also aims to assess the effect of TF-IDF method combined with IndoBERTweet and Word2Vect in classifying Indonesian language data. To be able to achieve these objectives, this research makes TF-IDF and IndoBERTweet as feature extraction and Word2Vect as feature expansion in the LSTM model. To fulfill the needs of the sentence-level granularity procedure, this research uses two types of data, namely manual labeling data and labeling system as a comparison of system performance in carrying out the task.

Further discussion in this paper will contain the following. Section 2 contains a description of the research method regarding sentence-level granularity-based sentiment analysis of Twitter comments. Section 3 contains results and discussion and followed by the conclusions drawn in Section 4.

2. METHODS

This research builds an LSTM system for sentence-level granularity-oriented sentiment analysis with TF-IDF feature extraction and IndoBERTweet Word Embedding as feature expansion. The system is used to classify comments with a dataset containing political, social, and economic issues in Indonesia language into three polarities (negative, neutral, and positive). Through the research conducted with system flow in Fig. 1, the best method in sentence-level granularity-oriented sentiment analysis of Indonesian language can be known over the evaluation step.

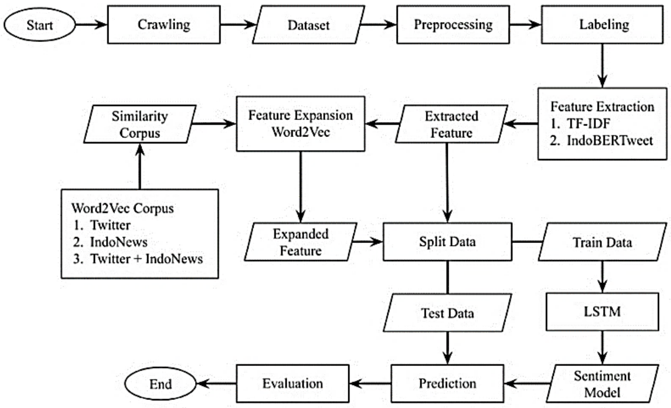

Fig. 1. Flowchart System

2.1. Sentiment Analysis

Sentiment analysis is gaining popularity due to the development of social media [14] which provides a place for users to express opinions and is a source of important elements in assessing a subject [15]. This procedure is an approach to natural language processing [14]. Sentiment analysis can classify text polarity into negative, neutral, and positive categories. This procedure is usually used to assess mass opinion by using computational tools on an object that can later be used for various purposes, including political, social, and economic [16].

In modeling sentiment analysis, this experiment uses deep learning which is capable of handling tasks with large amounts of high-dimensional data such as text data [17]. In addition, deep learning is also capable of processing combined data which is often referred as cross-modal learning [6],[18]. Cross-modal can handle the effect of context misalignment on related information by using context-gated to capture cross-modal interactions [19]. This process is useful in analyzing an application as a source of decision-making and providing recommendations based on information or data that has been processed [6]. Deep learning uses multilayer processing techniques that represent input data into numerical formulas to be classified at each layer [20].

2.2. Crawling

The dataset consists of Twitter user comments taken from Twitter Developer by using API Key and Consumer Key to retrieve data [21]. This research takes 50,000 tweet data containing keywords to be used as a dataset that represents sentiment based on political, social, and economic topics. Dataset reduction occurred due to data duplication, resulting in 30,811 tweets as shown in Table 1 being the final dataset before entering the preprocessing stage.

Table 1. Data Distribution by Keyword

Keyword	Amount
Polisi or Polri	2,191
Tentara or TNI	1,471
Presiden	2,194
Masyarakat	2,457
Pertamina	8,492
Pertalite	583
Bansos	10,000
BPJS	10,135
Total	30,811

2.3. Preprocessing

Text preprocessing processes raw text input data to be returned into word tokens according to a language structure that can be understood by the system. Tokens are single words or groups of words that are counted based on their frequency and serve as analysis features [22]. To be able to generate tokens, the dataset must go through the following process:

1. Cleaning removes characters that are not needed by the system by removing numbers, symbols, punctuation marks, links, and mentions.
2. Case folding converts all characters from the cleaning stage into a sentence format with lowercase letters.
3. Filtering is divided into two parts, namely Stopwords removal which removes connecting words such as "dan", "lalu", "yang", and others. Next, the process will go through the stage of converting slang words into normalization.
4. Stemming removes affixes and converts words into their original form.
5. Tokenizing is the last stage that breaks the sentence into word tokens in the form of an array.

Data must go through a preprocessing stage before being used by the system because preprocessing can affect the reliability and validity of the system's work based on the modeling results carried out [23].

2.4. Labeling

This stage is done by manually labeling [24] the dataset using a value scale (-1, 0, 1) to represent negative, neutral, and positive sentences. To ensure the correctness of the labeling results, at least each data is checked by three people. This labeling uses the majority vote [25] method for decision-making when there is a difference of opinion. The manual labeling results are presented in Table 3 and Table 4 which will be compared with the labeling system results.

In order to meet the needs of granularity-oriented sentiment analysis, data that is labeled by the system is created with the steps in Fig. 2. The beginning of the labeling process uses TF-IDF word relevance [26] to documents containing negative, neutral, and positive polarity taken from manual labeling data. This method is used to form a feature vector that compares three documents to evaluate how important is the word in the document [27]. When running TF-IDF, we also set the word length using the N-Gram range (1-3) to increase the understanding of the model in assessing the features used [28].

The 900 data with the highest TF-IDF value in each class were collected for further labeling through a survey. This stage was conducted by 30 people to assess the granularity weight of the word on a numerical scale from -5 to 5 to represent the "sense of the word" in each class as in Table 2. The labeling system was carried out by looking at the weight of the word referring to the survey data and added 10,320 data obtained from GitHub to build a corpus of data of sufficient granularity to produce labeled data as in Table 3 and Table 4.

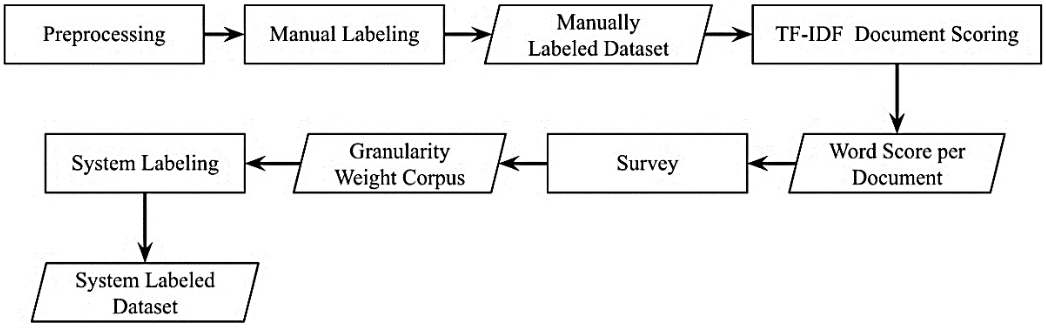

Fig. 2. Labeling Workflow

Table 2. Granularity Weight

Word	Negative (%)	Neutral (%)	Positive (%)	Weight
bakti	0	13	87	5
tenang bpjs	7	43	50	3
kasihan rakyat	37	37	27	-2
visi misi kabinet	0	70	30	0
sogok pakai bansos	70	27	3	-4

Table 3. Labeling Result

Tweet	Manual Labeling	System Labeling
debus kedok tentara	-1	0
bbm subsidi supply jual spbu asing lantas rugi pertamina	0	-1
naik bbm subsidi bansos tuju tekan inflasi my pertamina anggap solusi ketidaktepatan sasar	1	1

Table 4. Label Data Distribution

Label	Total	
	Manual Labeling	System Labeling
Negative	10,804	7,736
Neutral	9,591	3,579
Positive	10,416	19,496

2.5. Balancing Data

Data balancing is done so that the system can perform equally in every class [29]. The distribution of labeling system data is considered necessary for balancing so that the amount of data from each class is not much different. Random Undersampling and Random Oversampling (RUS-ROS) method are performed which selects data randomly [30]. RUS works by reducing the quantity of the majority class until it reaches the target data. While ROS works to increase the amount of minority data by duplicating data [31]. Researchers target each class to have 10,000 data. Therefore, RUS is applied to positive sentiment data by reducing 9,496 data and ROS is applied to neutral sentiment data by adding 6,421 data and 2,264 data for negative sentiment so that the target data is reached.

2.6. Feature Extraction

In text classification, the system must be able to classify the input data into predefined categories. To perform this task, a feature extraction stage is required so that the system can extract information and represent it based on the input data. Feature extraction forms an N-dimensional vector space, where each dimension represents one feature obtained from the dataset [32]. This research uses TF-IDF and IndoBERTweet feature extraction on the entire dataset by requiring feature selection to seek the number of features used to get optimal performance.

TF-IDF is a feature extraction approach that performs the weighting of word features in a simply and effectively. This method produces a high-dimensional corpus matrix, which can increase the possibility of overfitting the model. However, the problem can be overcome by reducing the word features used. The word feature weighting scheme is done by calculating the value of the feature by looking at the occurrence of the feature in the document [33].

$$w_{i,j} = tf_i \times \log\left(\frac{N}{df_i}\right) \tag{1}$$

TF-IDF calculation is done using the formula in (1) where $w_{i,j}$ is the weight of term i in document j. N represents the number of documents in the corpus, $d_{fi,j}$ is the term frequency of term i in document j, and d_{fi} is the frequency value of term i that occurs in the corpus [33].

Meanwhile, IndoBERTweet is a derivative of the Bidirectional Encoder Representations from Transformers (BERT) model as a model that uses Indonesian vocabulary obtained from Twitter. This method has been trained using 26 million tweets and 409 million words consisting of four main topics, namely, economy, health, education, and politics [13]. BERT performs embedding by reading input from two directions (from the left and right) from the corpus using 768 layers [6]. As feature extraction, BERT uses an attention model that works as embedding so that it can perform value weighting and classify text data into negative ($x < -0.01$), neutral ($-0.01 < x < 0.01$), and positive ($x > 0.01$) [34].

$$E_{IBT}(x) = \frac{1}{|T_{IB}(x)|} \sum_{y \in T_{\mathrm{IB}}(x)} E_{IB}(y) \tag{2}$$

Equation (2) is the IndoBERTweet formula to performing embedding where $T_{IB}(x)$ is the set of WordPiece tokens for word X produced by the IndoBERT tokenizer [33].

2.7. Feature Expansion

Feature Expansion method in this research is carried out using Word2Vec to reduce the possibility of vocabulary mismatches in the system [35]. Word2Vec generates word embedding from text data using Continous Bag of Bord (CBOW) and skip-gram approaches to find word similarity values from the available corpus [36]. This research collects Tweet and IndoNews data with data distribution as in Table 5 to build corpus similarity. Corpus building is done by assessing the context of words in pairs to pay attention to how often the word pairs appear [37].

Table 5. Word2Vec Corpus Data Distribution

Data	Amount
Tweet	12,000
IndoNews	20,000
Tweet + IndoNews	24,000

Table 6. Word Similarity of Polisi

Word	Top Similarity Rank				
	Top 1	Top 2	Top 3	Top 4	Top 5
Polisi	Oknum 83.81%	Bunuh 82.85%	Tangkap 81.20%	Aparat 80%	Anggota 77.98%
	Top 6	Top 7	Top 8	Top 9	Top 10
	Langgar 77.66%	Institusi 77.59%	Tegak 77.37%	Libat 76.97%	Hukum 76.91%

2.8. Modeling

This research builds a granularity-oriented sentiment analysis system using an LSTM model. The mentioned model is used because it is a deep learning model that works with an attention-based system that supports aspect-based oriented sentiment analysis procedures as it can focus on sentence-level analysis [2]. In addition, the model is considered superior to the Recurrent Neural Network (RNN) because it is able to handle the vanishing gradient problem by using gates that shows in (3)-(8) [39], [40] when processing long-sentence data such as text data, making the LSTM able to capture text dependencies on long-sentence input data [41]. The model must be able to classify Indonesian input data into negative, neutral, and positive sentiment groups. Before going through modeling, the dataset will be split into train data and test data with a ratio of 90:10, 80:20, and 70:30 [42] as a baseline determination. The next step is to combine feature extraction using TF-IDF and IndoBERTweet with feature expansion using Word2Vec following the scenario to strive for optimal modeling.

$$forget\ gate\ (f_t) = \sigma_g\ (W_f\ \times\ x_t + U_f\ \times\ h_{t-1} + b_f) \tag{3}$$

$$input\ gate\ \ (i_t) = \sigma_g\ (W_i\ \times\ x_t + U_i\ \times\ h_{t-1} + b_i) \tag{4}$$

$$output\ gate\ (o_t) = \sigma_g\ (W_o\ \times\ x_t + U_o\ \times\ h_{t-1} + b_o) \tag{5}$$

$$cell\ memory\ (c'_t) = \ \sigma_g\ (W_c\ \times\ x_t + U_c\ \times\ h_{t-1} + b_c) \tag{6}$$

$$cell\ state\ (c_t) = \ f_t\ \times\ c_{t-1} + i_t\ \times\ c'_t \tag{7}$$

$$Hidden\ state\ (h_t) = \ o_t\ \times\ \sigma_c(c_t) \tag{8}$$

2.9. Performance Evaluation

The performance of the model will be evaluated using Multiclass Confusion Matrix which groups 3 classes eventually while considering precision, recall, f1 score, and accuracy matrices that shows in equation (9)-(11) [43]. The f1 score matrix is used to assess the predictive balance of the precision and recall matrices. Meanwhile, the accuracy matrix is used to assess the classification performance of the model [44].

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$F1\ Score = \ 2\ \times \frac{Precision\ \times\ Recall}{Precision\ +\ Recall} \tag{10}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

TP is known as true positive, *TN* as true negative, *FP* as false positive, and *FN* as false negative which represents the predicted value with the true value.

3. RESULTS AND DISCUSSION

In this test, 4 scenarios are made that make up the system as in Table 7. Each scenario will compare the accuracy of the methods used along with the comparison of accuracy against manual labeling datasets and system labeling data based on granularity weights.

Table 7. Model Scenario

Skenario	Model
1	LSTM + Ngram (Baseline)
2	Baseline + TF-IDF
3	Baseline + TF-IDF + IndoBERTweet
4	Baseline + TF-IDF + IndoBERTweet + Word2Vec

3.1. Scenario 1

Scenario 1 was conducted by searching for a baseline against the dataset by trying 3 data splitting ratios namely 90:10, 80:20, and 70:30. The baseline search also includes word-sequence search using N-grams with the range of n used is 1-gram, 2-gram, 3-gram, (1-2)-gram, and (1-3)-gram. The baseline will be selected based on the highest accuracy value by looking at the ratio size and word-sequence type. Manual labeling data that has been split will directly enter the modeling stage. Meanwhile, the labeling system data will go through a balancing process with the RUS-ROS method first. This is done to reduce the possibility of overfitting because the model is only able to learn the majority class due to the large data difference with the minority class.

After obtaining the test results in Table 7, the baseline for manual labeling data was selected with a split data ratio of 90% for train data and 10% for test data with 1-gram word-sequence because it has the highest accuracy value. The baseline selection for system labeling data is only done in 1-gram, (1-2)-gram and (1-3)-gram word-sequences because it refers to the results of Table 8. The results of testing the baseline data of the labeling system in Table 9 show that the highest accuracy is in the 90:10 ratio data with 1-gram data.

Table 8. Manual Labeling Baseline Testing Results

Ratio	**Metrics**	**LSTM + 1-gram**	**LSTM + 2-gram**	**LSTM + 3-gram**	**LSTM + (1-2)-gram**	**LSTM + (1-3)-gram**
90:10	Accuracy (%)	**68.27**	67.06	65.81	67.47	66.73
	F1-Score (%)	**68.37**	67.08	65.82	67.37	66.63
80:20	Accuracy (%)	66.86	66.18	65.38	67.31	65.83
	F1-Score (%)	66.87	66.15	65.42	67.31	65.88
70:30	Accuracy (%)	66.66	65.76	64.63	66.72	65.42
	F1-Score (%)	66.64	65.94	64.59	66.53	65.51

Table 9. System Labeling Baseline Testing Results

Ratio	**Metrics**	**LSTM + 1-gram**	**LSTM + (1-2)-gram**	**LSTM + (1-3)-gram**
90:10	Accuracy (%)	**87.38**	86.70	85.52
	F1-Score (%)	**87.32**	86.48	85.17
80:20	Accuracy (%)	86.63	85.70	84.62
	F1-Score (%)	86.43	85.38	84.67
70:30	Accuracy (%)	85.48	85.21	83.84
	F1-Score (%)	85.39	84.96	83.64

3.2. Scenario 2

In scenario 2, modeling will use the TF-IDF method on the entire dataset. To add TF-IDF method, it is necessary to do feature selection by testing the number of features to be used in the model in order to get optimal results. Based on Table 10, it can be concluded that the max feature with the highest accuracy is 8,000 for manual labeling data and 10,000 for system labeling data which will then become the default value in every test using TF-IDF. From the test results, it is found that the use of the TF-IDF method increases accuracy by 3.7% for manual labeling data and 7.1% for system labeling data so that the TF-IDF method will always be used in scenarios 3 and 4.

Table 10. Max Feature in Scenario 2 Testing Result

Max Feature	**Manual Labeling**		**System Labeling**	
	Accuracy (%)	F1-Score (%)	Accuracy (%)	F1-Score (%)
1000	68.20	68.14	82.57	82.58
3000	69.57	69.51	91.17	91.18
5000	70.08	70.19	91.57	91.56
8000	**70.85 (+3.7%)**	**70.89 (+3.6%)**	92.87	92.87
10000	68.95	68.99	**93.67 (+7.1%)**	**93.46 (+7.0%)**
10755	69.24	69.29	93.33	93.33

3.3. Scenario 3

This test compares the accuracy of baseline with the accuracy of adding the IndoBERTweet method. Table 11 shows that the addition of IndoBERTweet method can increase the accuracy of the model by 8.5% for manual labeling data and 7.2% for system labeling data. However, the addition of the method also resulted decrease in F1-Score on manual labeling data by -0.3%.

Table 11. Scenario 3 Testing Result

Model	Manual Labeling		System Labeling	
	Accuracy (%)	F1-Score (%)	Accuracy (%)	F1-Score (%)
Baseline	68.27	68.37	87.38	87.32
Baseline + TF-IDF + IndoBERTweet	**74.11 (+8.5%)**	68.13 (-0.3%)	**93.80 (+7.3%)**	**93.68 (+7.2%)**

3.4. Scenario 4

Since the addition of IndoBERTweet method improves the accuracy of the model, it will be used in scenario 4. This scenario will use feature expansion with Word2Vec method which will be combined with TF-IDF and IndoBERTweet methods. The addition of this feature expansion uses the corpus word similarity that has been built previously. In addition, it is necessary to test to determine the top rank word similarity that will be used in the feature expansion method.

The test results in Table 12 show that the model accuracy will be optimal by using top 1 similarity with the Twitter + IndoNews corpus for manual labeling data and top 1 for Twitter corpus data against system labeling data. The use of Word2Vect feature expansion provides a significant improvement for manual labeling data which is 30.3%. Meanwhile, in the system labeling data, the method can increase the model accuracy by 11.9%.

Tabel 12. Scenario 4 Testing Results

Corpus	Similarity Rank	Manual Labeling		System Labeling	
		Accuracy (%)	F1-Score (%)	Accuracy (%)	F1-Score (%)
Twitter	Top 1	80.34	78.26	**97.80 (+11.9%)**	**97.80 (+12%)**
	Top 5	85.79	85.07	97.77	97.76
	Top 10	82.61	81.27	97.00	97.00
IndoNews	Top 1	81.57	80.93	97.67	97.66
	Top 5	84.94	84.03	93.20	93.18
	Top 10	82.35	80.69	96.27	96.25
Twitter + IndoNews	Top 1	**88.97 (+30.3%)**	**88.89 (+30%)**	97.03	97.04
	Top 5	85.72	85.19	97.70	97.70
	Top 10	87.48	87.49	96.67	96.68

3.5. Result Analysis

Through the results of model testing using 4 scenarios, it can be seen that the addition of feature extraction and feature expansion methods can improve model accuracy. As presented in Fig. 3 and Fig. 4, we can conclude that the classification of opinion sentences by looking at the polarity of responses into negative, neutral, and positive sentiments using sentence-level granularity sentiment analysis procedures with an accuracy of 97.80% can outperform conventional sentiment analysis procedures with an accuracy value of 88.97%. With these test results, this research is also said to outperform the research conducted by Ling [10] who also conducted granularity-oriented sentiment analysis combining C-LSTM, TF-IDF, and ELMo with a final result of 81.31%.

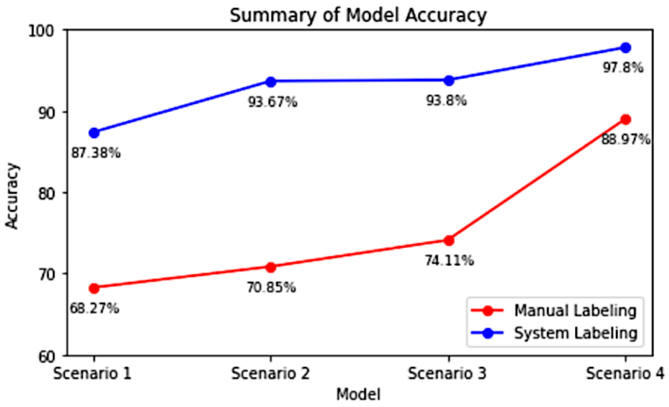

Fig. 3. Model Accuracy per Scenario

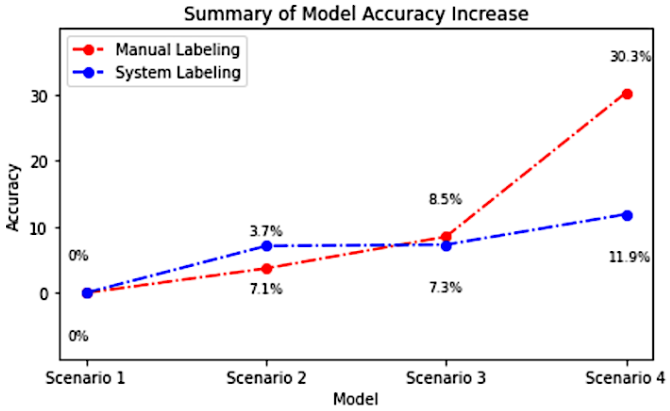

Fig. 4. Model Accuracy Improvement

The use of TF-IDF and IndoBERTweet feature extraction can increase model performance by increasing accuracy by 7.3% for manual labeling and 8.5% for the labeling system against the baseline. Meanwhile, the largest increase in accuracy occurred due to the use of Word2Vec by 30.3% and 11.9% using Twitter and IndoNews corpus to build word semantics. The increase in accuracy after using these methods can certainly answer the effectiveness of the method against the model built so as to produce an optimal system.

Based on the results of this research, it can be said that this research has succeeded in creating an optimal system to perform the task of classifying social media data on Indonesian opinion sentences. After doing research, researchers have not found research that combines LSTM, TFIDF, IndoBERTweet, and Word2Vec models for sentence-level granularity-oriented sentiment analysis like this.

4. CONCLUSION

This research was conducted to create a system that is able to classify Indonesian input data into negative, neutral, and positive polarity. The construction of this system uses LSTM as a granularity-oriented sentiment analysis model to be able to handle long and detailed sequence data. The dataset used amounted to 30,811 Indonesian tweets with political, social, and economic topics. The dataset is labeled in two ways, namely manually labeling and granularity system labeling. To produce an optimal model, feature extraction is carried out using TF-IDF and IndoBERTweet as word weighting. In addition, Word2Vec formed from Twitter and IndoNews corpus is used as feature expansion so that it can identify semantic words. Based on the test results, the addition of these methods is considered appropriate because it can significantly increase the accuracy of the model, which is 88.97% for manual labeling data and 97.80% for system labeling data. This shows that Indonesian Twitter classification works better when using a sentence-level granularity-oriented sentiment analysis system. Therefore, as a suggestion for further research, the researcher recommends trying to combine these methods by using optimization algorithms to find out the use of these methods to increase the accuracy of the system.

REFERENCES

[1] A. Kumar and A. Jaiswal, “Systematic literature review of sentiment analysis on Twitter using soft computing techniques,” in *Concurrency and Computation: Practice and Experience*, vol. 32, no. 1, 2020, https://doi.org/10.1002/cpe.5107.

[2] A. Yadav and D. K. Vishwakarma, “Sentiment analysis using deep learning architectures: a review,” *Artif Intell Rev*, vol. 53, no. 6, pp. 4335–4385, 2020, https://doi.org/10.1007/s10462-019-09794-5.

[3] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, “A survey of sentiment analysis in social media,” *Knowl Inf Syst*, vol. 60, no. 2, pp. 617–663, 2019, https://doi.org/10.1007/s10115-018-1236-4.

[4] Z. Drus and H. Khalid, “Sentiment analysis in social media and its application: Systematic literature review,” in *Procedia Computer Science*, vol. 161, pp. 707–714, 2019, https://doi.org/10.37943/AITU.2021.57.68.005.

[5] A. Mukasheva, “Tasks and Methods of Text Sentiment Analysis,” *Scientific Journal of Astana IT University*, no. 7, pp. 55–62, 2021, https://doi.org/10.37943/AITU.2021.57.68.005.

[6] C. Janiesch, P. Zschech, and K. Heinrich, “Machine learning and deep learning,” *Electronic Markets*, vol. 31, no. 3, pp. 685-695, 2021, https://doi.org/10.1007/s12525-021-00475-2.

[7] R. K. Dey, D. Sarddar, I. Sarkar, R. Bose and S. Roy, “A Literature Survey on Sentiment Analysis Techniques involving Social Media and Online Platforms,” *International Journal Of Scientific & Technology Research*, vol. 1, no. 1, pp. 166-173, 2020, http://www.ijstr.org/final-print/may2020/A-Literature-Survey-On-Sentiment-Analysis-Techniques-Involving-Social-Media-And-Online-Platforms.pdf.

[8] M. Birjali, M. Kasri, and A. Beni-Hssane, “A comprehensive survey on sentiment analysis: Approaches, challenges and trends,” *Knowl Based Syst*, vol. 226, 2021, https://doi.org/10.1016/j.knosys.2021.107134.

[9] N. C. Dang, M. N. Moreno-García, and F. de la Prieta, “Sentiment analysis based on deep learning: A comparative study,” *Electronics (Switzerland)*, vol. 9, no. 3, 2020, https://doi.org/10.3390/electronics9030483.

[10] M. Ling, Q. Chen, Q. Sun, and Y. Jia, “Hybrid Neural Network for Sina Weibo Sentiment Analysis,” *IEEE Trans Comput Soc Syst*, vol. 7, no. 4, pp. 983–990, 2020, https://doi.org/10.1109/TCSS.2020.2998092.

[11] Q. Bai, J. Zhou, and L. He, “PG-RNN: using position-gated recurrent neural networks for aspect-based sentiment classification,” *Journal of Supercomputing*, vol. 78, no. 3, pp. 4073–4094, 2022, https://doi.org/10.1007/s11227-021-04019-5.

[12] H. Fan *et al.*, “Social media toxicity classification using deep learning: Real-world application uk brexit,” *Electronics (Switzerland)*, vol. 10, no. 11, 2021, https://doi.org/10.3390/electronics10111332.

[13] F. Koto, J. H. Lau, and T. Baldwin, “IndoBERTweet: A Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Vocabulary Initialization,” Sep. 2021, https://doi.org/10.18653/v1/2021.emnlp-main.833.

[14] S. Kamiş and D. Goularas, “Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data,” in *Proceedings - 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications, Deep-ML 2019*, pp. 12–17, 2019, https://doi.org/10.1109/Deep-ML.2019.00011.

[15] N. C. Dang, M. N. Moreno-García, and F. de la Prieta, “Sentiment analysis based on deep learning: A comparative study,” *Electronics (Switzerland)*, vol. 9, no. 3, 2020, https://doi.org/10.3390/electronics9030483.

[16] K. Chakraborty, S. Bhattacharyya, and R. Bag, “A Survey of Sentiment Analysis from Social Media Data,” *IEEE Trans Comput Soc Syst*, vol. 7, no. 2, pp. 450–464, 2020, https://doi.org/10.1109/TCSS.2019.2956957.

[17] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, “Deep Learning-Based Text Classification,” *ACM Computing Surveys*, vol. 54, no. 3, 2021, https://doi.org/10.1145/3439726.

[18] L. Zhen, P. Hu, X. Wang, and D. Peng, “Deep Supervised Cross-modal Retrieval,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10394-10403, 2019, https://www.computer.org/csdl/proceedings-article/cvpr/2019/329300k0386/1gyro4RRiJG.

[19] H. Wen, S. You, and Y. Fu, “Cross-modal context-gated convolution for multi-modal sentiment analysis,” *Pattern Recognit Lett*, vol. 146, pp. 252–259, 2021, https://doi.org/10.1016/j.patrec.2021.03.025.

[20] H. H. Do, P. W. C. Prasad, A. Maag, and A. Alsadoon, “Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review,” *Expert Systems with Applications*, vol. 118, pp. 272–299, 2019, https://doi.org/10.1016/j.eswa.2018.10.003.

[21] J. G. D. Harb, R. Ebeling, and K. Becker, “A framework to analyze the emotional reactions to mass violent events on Twitter and influential factors,” *Inf Process Manag*, vol. 57, no. 6, 2020, https://doi.org/10.1016/j.ipm.2020.102372.

[22] M. Anandarajan, C. Hill, and T. Nolan, “Text Preprocessing,” *Practical text analytics: Maximizing the value of text data*, pp. 45–59, 2019, https://doi.org/10.1007/978-3-319-95663-3_4.

[23] L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, “Text Preprocessing for Text Mining in Organizational Research: Review and Recommendations,” *Organ Res Methods*, vol. 25, no. 1, pp. 114–146, 2022, https://doi.org/10.1177/1094428120971683.

[24] F. Gargiulo, S. Silvestri, M. Ciampi, and G. de Pietro, “Deep neural network for hierarchical extreme multi-label text classification,” *Applied Soft Computing Journal*, vol. 79, pp. 125–138, 2019, https://doi.org/10.1016/j.asoc.2019.03.041.

[25] J. Hartmann, J. Huppertz, C. Schamp, and M. Heitmann, “Comparing automated text classification methods,” *International Journal of Research in Marketing*, vol. 36, no. 1, pp. 20–38, 2019, https://doi.org/10.1016/j.ijresmar.2018.09.009.

[26] Y. Zhang, Y. Zhou, and J. T. Yao, “Feature Extraction with TF-IDF and Game-Theoretic Shadowed Sets,” in *Communications in Computer and Information Science*, vol. 1237, pp. 722–733, 2020, https://doi.org/10.1007/978-3-030-50146-4_53.

[27] M. Umadevi, “Document Comparison Based on TF-IDF Metric,” *International Research Journal of Engineering and Technology*, vol. 2, pp. 1546-1550, 2020, [Online]. Available: www.irjet.net.

[28] S. Wattanakriengkrai *et al.*, "Automatic Classifying Self-Admitted Technical Debt Using N-Gram IDF," *2019 26th Asia-Pacific Software Engineering Conference (APSEC)*, pp. 316-322, 2019, https://doi.org/10.1109/APSEC48747.2019.00050.

[29] M. A. Jamal, M. Brown, M.-H. Yang, L. Wang, and B. Gong, “Rethinking Class-Balanced Methods for Long-Tailed Visual Recognition from a Domain Adaptation Perspective,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7610-7619, 2020, https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.00763.

[30] A. I. S. Aftab and F. Matloob, “Performance Analysis of Resampling Techniques on Class Imbalance Issue in Software Defect Prediction,” *International Journal of Information Technology and Computer Science*, vol. 11, no. 11, pp. 44–53, 2019, https://doi.org/10.5815/ijitcs.2019.11.05.

[31] F. Alahmari, “A Comparison of Resampling Techniques for Medical Data Using Machine Learning,” *Journal of Information and Knowledge Management*, vol. 19, no. 1, 2020, https://doi.org/10.1142/S021964922040016X.

[32] M. Y. V. Nagessh and T. Anuradha, “A Word2Vector Representation for Twitter Sentimental Analysis,” 2019, [Online]. Available: www.joics.org .

[33] R. Dzisevič and D. Šešok, "Text Classification using Different Feature Extraction Approaches," *2019 Open Conference of Electrical, Electronic and Information Sciences (eStream)*, pp. 1-4, 2019, https://doi.org/10.1109/eStream.2019.8732167.

[34] M. Hoang, O. A. Bihorac, and J. Rouces, “Aspect-Based Sentiment Analysis Using BERT,” in *Proceedings of the 22nd nordic conference on computational linguistics*, pp. 187-196, 2019, https://aclanthology.org/W19-6120.

[35] A. R. Royyan and E. B. Setiawan, “Feature Expansion Word2Vec for Sentiment Analysis of Public Policy in Twitter,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 1, pp. 78–84, 2022, https://doi.org/10.29207/resti.v6i1.3525.

[36] H. A. Almuzaini and A. M. Azmi, "Impact of Stemming and Word Embedding on Deep Learning-Based Arabic Text Categorization," *in IEEE Access*, vol. 8, pp. 127913-127928, 2020, https://doi.org/10.1109/ACCESS.2020.3009217.

[37] N. A. Nugroho and E. B. Setiawan, “Implementation Word2Vec for Feature Expansion in Twitter Sentiment Analysis,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 5, pp. 837–842, 2021, https://doi.org/10.29207/resti.v5i5.3325.

[38] M. P. K. Dewi and E. B. Setiawan, “Feature Expansion Using Word2vec for Hate Speech Detection on Indonesian Twitter with Classification Using SVM and Random Forest,” *Jurnal Media Informatika Budidarma*, vol. 6, no. 2, p. 979, 2022, https://doi.org/10.30865/mib.v6i2.3855.

[39] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network,” *Physica D*, vol. 404, 2020, https://doi.org/10.1016/j.physd.2019.132306.

[40] J. Dai, C. Chen, and Y. Li, “A backdoor attack against LSTM-based text classification systems,” *IEEE Access*, vol. 7, pp. 138872–138878, 2019, https://doi.org/10.1109/ACCESS.2019.2941376.

[41] L. Khan, A. Amjad, K. M. Afaq, and H. T. Chang, “Deep Sentiment Analysis Using CNN-LSTM Architecture of English and Roman Urdu Text Shared in Social Media,” *Applied Sciences (Switzerland)*, vol. 12, no. 5, 2022, https://doi.org/10.3390/app12052694.

[42] A. M. U. D. Khanday, S. T. Rabani, Q. R. Khan, N. Rouf, and M. Mohi Ud Din, “Machine learning based approaches for detecting COVID-19 using clinical text data,” *International Journal of Information Technology (Singapore)*, vol. 12, no. 3, pp. 731–739, 2020, https://doi.org/10.1007/s41870-020-00495-9.

[43] I. Markoulidakis, G. Kopsiaftis, I. Rallis, and I. Georgoulas, “Multi-Class Confusion Matrix Reduction method and its application on Net Promoter Score classification problem,” in *ACM International Conference Proceeding Series*, pp. 412–419, 2021, https://doi.org/10.1145/3453892.3461323.

[44] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B. W. On, “Fake news stance detection using deep learning architecture (CNN-LSTM),” *IEEE Access*, vol. 8, pp. 156695–156706, 2020, https://doi.org/10.1109/ACCESS.2020.3019735.

BIOGRAPHY OF AUTHORS

Nisa Maulia Azahra is currently pursuing a bachelor's degree in computer science at Telkom University, Indonesia. She is very keen on the world of data science and won a national competition in the field in 2021. She continued to take part in training and started working as a data scientist in her final year of study. Email: mauliaazahranisa@student.telkomuniversity.ac.id

Erwin Budi Setiawan is a senior lecturer in School of Computing, Telkom University, Bandung, Indonesia. He has more than 10 years Research and Teaching experience in the domain of Informatics. Currently, he is a Associate Professor. His research interests are machine learning, people analytic, and social media analysis. Email: erwinbudisetiawan@telkomuniversity.ac.id