Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
14/1 (2021), 1-8. DOI: http://dx.doi.org/10.21609/jiki.v14i1.812

Sarcasm Detection Engine for Twitter Sentiment Analysis using Textual and Emoji Feature

Bagus Satria Wiguna, Cinthia Vairra Hudiyanti, Alqis Rausanfita, Agus Zainal Arifin, Rizka W. Sholikah

[1234] Department of Informatics, Faculty of Information and Communication Technology, Sepuluh Nopember Institute of Technology, Surabaya, 60111, Indonesia

E-mail: wiguna.bagussatria@gmail.com cvairra@gmail.com qisfit@gmail.com agusza@if.its.ac.id rizka.wakhidatus15@mhs.if.its.ac.id

Abstract

Twitter is a social media platform used to express sentiments about events, topics, individuals, and groups. Sentiments in Tweets can be classified as positive or negative expressions. However, the sentiment is an expression that is the opposite of what is meant to be, and this is called sarcasm. The existence of sarcasm in a Tweet is chalenging to be detected automatically by a system, even by humans. In this research, we propose a weighting scheme based on the inconsistency between the sentiment of Indonesian tweets and the usage of emoji. The weighting scheme for detecting sarcasm can be used to find out a sentiment about an event, topic, individual, group, or product's review. The proposed method calculates the distance between the textual feature polarity score obtained from the Convolutional Neural Network and the emoji polarity score in a Tweet. This method is used to find the boundary value between Tweets that contain sarcasm or not. The model's experimental results developed obtained an f1-score of 87.5%, precision 90.5%, and recall 84.8%.

Keywords: *twitter, sentiment analysis, sarcasm, social media, textual feature, emoji feature*

1. Introduction

One example of social media that is widely used today is Twitter. Twitter is a social media platform used to discuss sentiments about events, topics, individuals, and groups [1]. Sentiment analysis (opinion mining) techniques analyze opinionated text, which contains people’s opinions toward entities such as products, organizations, individuals, and events [2]. Sentiment snippets are an essential part of both companies and individuals looking to monitor their reputation [3]. They can be used as a convinient tool for feedback on their products and actions. The sentiment of tweets can be classified as positive responses or negative responses. Sentiments contained in tweets attract several companies or organizations, or individuals to dig up some information. because the number of characters that can be written in a tweet is limited, causing people to express their opinions using slang, characters, Etc., which sometimes the understanding of the use of these characters is not the same between people [4].

In a sentiment, some expressions contradict what they mean. The different meanings and expressions are called sarcasm [5]. The existence of sarcasm in tweets is challenging to detect automatically by a system, even by humans, because of textual data in tonal and genital instructions such as speech tone pressure, eye friction, hand movements, and whether it cannot be detected[6].

The content of tweets is textual features that contain sentences or words and non-textual features, namely emoji. When users write sarcasm expressions on tweets, they will deviate from the use of emoji. The positive sentiment of the tweet will be paired with negative value emojis and vice versa. Therefore the value of sarcasm in a tweet sentiment can be obtained based on sentiment analysis in the context of sentences and emojis in a tweet.

Sentiment analysis is a part of Natural Language Processing (NLP), which is related to finding the intention of opinions in a piece of text about the topic being discussed [6]. Sentiment analysis will identify sentiments in an expression, which then classifies based on its polarity score [7].

Several studies have been conducted to test sarcasm in textual data. Kumar's research [8] conducted sarcasm classification of a novel approach using the Content-Based Feature Selection Method. The data consists of an Amazon review. The feature selection stage is carried out in two stages. The first feature is selected using a comparison method between chi-square, information gain, and mutual information. In the second stage, the grouping is done to choose the features that best represent the Related features using the k-means algorithm. The next step is to compare the text classification results between the SVM method and the random forest method. The study [9] focused on the score to get the results of the detection of sarcasm. The recommended score is the sarcasm score obtained from the comparison of tweets with the corpus-based on sarcasm. In [8] and [9], sarcasm detection is based on textual data features that will get good results only if sentence content is long enough and tweets also contain short text. Therefore, we assume that sarcasm detection is difficult to deal with only with a sentiment in the text. Besides, research related to the detection of irregularities in Indonesian tweets is still rare. Therefore we focus on tweets in Indonesian.

In this research, we propose a weighting scheme based on the inconsistency between the sentiment of tweets in Indonesian and emoji usage. The proposed method calculates the distance between the polarity of textual features obtained from the convolutional neural network and the non-textual polarity score (emoji) in a tweet. The method is used to find the boundary value between tweets that contain sarcasm or not.

2. Methodology

In our proposed method, the model we build is used to detect sarcasm in tweets that can be done using two features, textual and non-textual features such as emojis. The two main features will be calculated based on the polarity score, then labeled positive, negative, and neutral. However, the neutral label is no longer needed because it does not effect on the other process.

After getting a label from each feature, the filtering is done to remove features with a neutral label. Then the value of two features in the tweet is compared. If one of the features is the opposite of the other features, then the tweet's sarcasm label is positive and vice versa.The tweet dataset that already has a label will be used as training data and testing data to build a sarcasm detection engine. The example of the dataset is shown in Table 1.

Table 1. Example dataset

TWEET
@JOKOWI AKU SETUJUU KOK PAK JOKOWI KLO IBUKOTA PINDAH.. KE PAPUA JUGA GA PAPA , ASYIK KAYKNYA 😁😁😁🙏🙏👍👍👍
@BPJSKESEHATANRI NGGAK ADA ISTILAH RUGI SELAMA PAK @JOKOWI YG JADI PRESIDEN.. LOVE JOKOWI ❤️❤️
WAHHH KOK PERWAKILAN LUAR NEGERI LEBIH SERING KE KANTORNYA PAK PRABOWO DARIPADA KE ISTANA?BINGUNG AKUTUHH 😭😭 HTTPS://T.CO/SO9G6FJT6X

The sarcasm detection engine has two main components, and the first is a text sentiment classifier using CNN and Emoji sentiment classifier. Input for the text sentiment classifier is text features from training and testing dataset, and the training dataset is used to train the CNN and testing dataset to get the sentiment score.

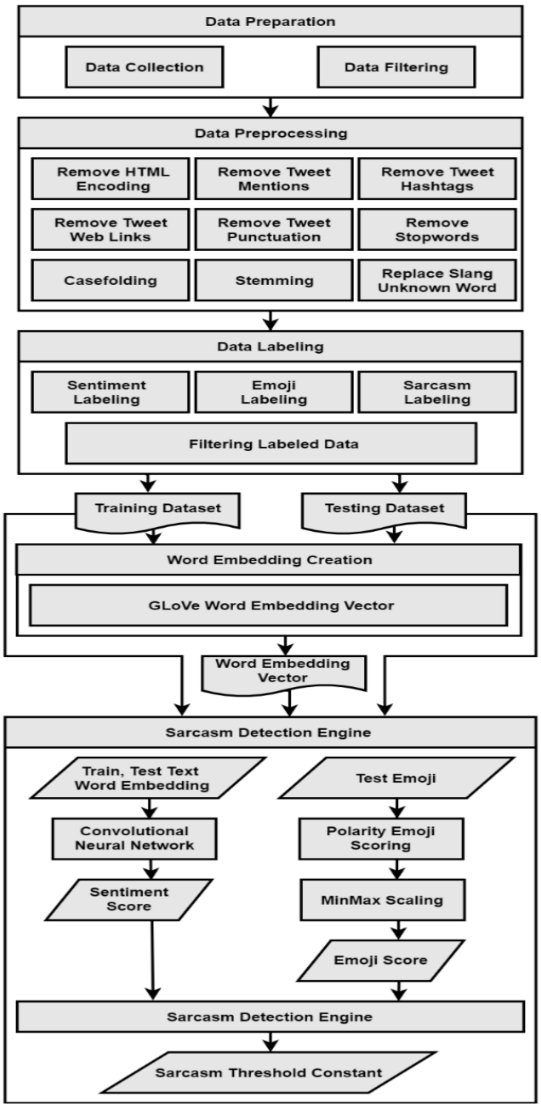


Figure 1. Proposed method

The emoji sentiment classifier's input is the emoji feature of the testing dataset to calculate the polarity score of the testing dataset. After getting the sentiment value of each feature from the testing dataset, the sarcasm classification calculates the difference in distance from the text and emoji features. The determination of the optimal threshold for sarcasm labeling is obtained from the f1-score of the predetermined interval.

Figure 1 explains the proposed method's steps, and each step will be explained more in the explanation.

2.1. Data Preparation

At the stage of data preparation, Twitter data retrieval is taken from 3 November 2019 to 10 November 2019. In this research, we can only retrieve data within seven days due to the unpaid Twitter public fire limitations. During the research process, political topics became trending topics, so we only used political topics where at that time, the political topics had many controversial things made/taken up by political figures. The keywords we use to collect tweets are 'Jokowi', 'Prabowo', 'Fachrul Razi', and 'Anis Baswedan'.

Table 2. Result of raw data crawling

KEYWORD	QUANTITY
JOKOWI	53661
PRABOWO	22492
FACHRUL RAZI	1374
ANIS BASWEDAN	497

Table 2 explains the raw data obtained for each keyword, and the total obtained tweets are 77961. At the filtering stage, the filtering of the tweets is already obtained. At the filtering stage, filtering will be done by removing a tweet containing emojis automatically using dictionary emojis [10]. The total dataset [1]is 6478 tweets.

2.2. Data Preprocessing

The tweet data that has been obtained needs to be done by preprocessing data. This research's preprocessing data stages by removing HTML encoding, mentions, hashtags, weblinks, punctuation, and stopwords. After that, case folding, stemming, and replacing slang & unknown words are applied for each word in the tweet. Preprocessing data needs to be done because the tweet data is unstructured, and there is noise.

Stopword deletion needs to be done to eliminate words that have no meaning. The stopword dictionary used comes from the NLTK and Sastrawi libraries. Stemming is used to change words into basic words by removing suffix, infix, prefix, and confix affixes.

Table 3. Sample of slang and unknown word

WORD	MEANING	IN ENGLISH
SAPATAU	SIAPA TAU	WHO KNOWS
CAWE	PANGGIL	CALL
KAGA	TIDAK	NO
VERY	SANGAT	VERY
INSHALAH	INSYAALLAH	IF ALLAH WILLS
GN	TIDAK	NO
FUCKING	SIAL	FUCKING
UDEH	UDAH	DONE
ORGX	ORANG	PEOPLE
MABUEK	MABUK	DRUNK
NGUPI	MINUM KOPI	DIRNK A COFFE

Replacement of slang and unknown words is done by building a custom slang and an unknown dictionary. The slang and unknown word dictionary are obtained from searching every word in the dataset into Kamus Besar Bahasa Indonesia (KBBI). However, If the word is not in the KBBI, it is a candidate for the slang / unknown word. Making a dictionary of the words dictionary is done manually annotated. Table 3 is a sample from the slang and unknown word dictionary. In this study, there were 6306 slang words and unknown words.

2.3. Data Sentiment Labeling

At the data labeling stage, the preprocessed dataset will be labeled. Each tweet contained in the dataset has three labels, namely sentiment label, emoji label, and sarcasm label.

Sentiment labeling is done by using the SentiWord dictionary. SentiWord is a lexicon-based sentiment feature that is generally used for sentiment analysis, and SentiWord deriving a high precision and high coverage lexicon for sentiment analysis [13]. The SentiWord dictionary is built from a collection of positive, negative, and neutral values. In this research, we use Barasa SentiWord[2], which belongs to David Moeljadi, to label sentiment value from a tweet.

$$pos\_ratio_t = \frac{\sum positive\_word_t}{total\_token_t} \tag{1}$$

$$neg\_ratio_t = \frac{\sum Negative\_word_t}{total\_token_t} \tag{2}$$

$$\begin{array}{c} if\ pos\_ratio_t > neg\_ratio_t\ , then\ positive\ or\ 1 \\ if\ pos\_ratio_t > neg\_ratio_t\ , then\ negative\ or\ 0 \\ if\ pos\_ratio_t = neg\_ratio_t\ , then\ neutral\ or\ 2 \end{array} \tag{3}$$

[1] https://intip.in/SRCSMP

[2] https://github.com/neocl/barasa

Equation 3 is the rule for the sentiment label of a tweet. In equation 1 is a positive ratio value obtained from the number of positive words in a tweet divided by the total words in the tweet. In equation 2, a negative ratio value is obtained from the number of negative words in a tweet divided by the number of words in a tweet. The words used in SentiWord are a type of noun, verb, adverb, and adjective.

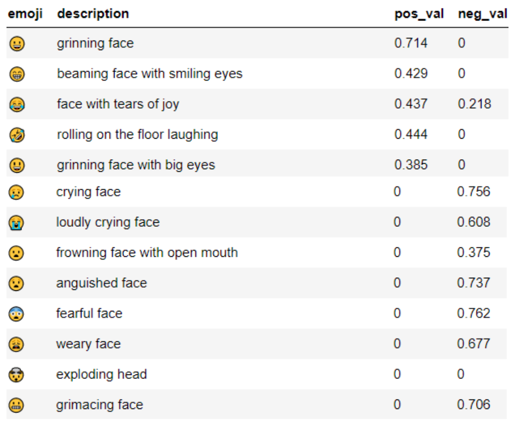
Table 4. Sentiment labeling result

SENTIMENT LABEL	QUANTITY
POSITIVE	1715
NEGATIVE	2763
NEUTRAL	2000

The results of sentiment labeling are obtained in Table 4. After getting the sentiment label, the filtering dataset is done by removing tweets with a neutral value of sentiment label.

2.4. Data Emoji Labeling

Emojis are graphical representations of user feelings. Emojis are generally in the form of character combinations or Unicode. Emojis are very effective in describing the condition of one's feelings[10].



emoji	description	pos_val	neg_val
😀	grinning face	0.714	0
😁	beaming face with smiling eyes	0.429	0
😂	face with tears of joy	0.437	0.218
🤣	rolling on the floor laughing	0.444	0
😃	grinning face with big eyes	0.385	0
😢	crying face	0	0.756
😭	loudly crying face	0	0.608
😦	frowning face with open mouth	0	0.375
😧	anguished face	0	0.737
😨	fearful face	0	0.762
😩	weary face	0	0.677
🤯	exploding head	0	0
😬	grimacing face	0	0.706

Figure 2. Emoji polarity lexicon

Emoji labeling is done using the emoji polarity dictionary[14]. In the emoji polarity dictionary, there are positive and negative polarity values for each emoji. Emoji labeling is explained in equation 4 below.

$$\begin{aligned} &if\ pos\_count_t > neg\_count_t\ , then\ positive\ or\ 1 \\ &if\ pos\_count_t < neg\_count_t\ , then\ negative\ or\ 0 \\ &if\ pos\_count_t = neg\_count_t\ , then\ neutral\ or\ 2 \end{aligned} \tag{4}$$

Where pos_count_t is the number of positive-value emojis while neg_count_t is the number of negative-value emojis.

Table 5. Emoji labeling result

EMOJI LABEL	QUANTITY
POSITIVE	3045
NEGATIVE	264
NEUTRAL	1169

Table 5 shows the results of emoji labeling. Tweets that have a neutral label emoji will be discarded.

2.5. Sarcasm Sentiment Labeling

The sarcasm sentiment labeling stage is the last step in the data labeling step. Sarcasm labeling is done by using the rules described in equation 5 below.

$$\begin{aligned} &if\ sentiment\_label_t = emoji\_label_t, then\ positive\ or\ 1 \\ &if\ sentiment\_label_t \neq emoji\_label_t, then\ negative\ or\ 0 \end{aligned} \tag{5}$$

In equation 5, when the value of being different with the value of then the value of Sarcasm label is positive, if the value of the two labels is the same, the value of the Sarcasm label is negative. A tweet is called positive sarcasm if there is a deviation from emojis from a sentence in a tweet or vice versa, but a tweet can be called negative sarcasm if the use of emojis matches the sentence conveyed in the tweet. [15]

Table 6. Sarcasm labeling result

SARCASM LABEL	QUANTITY
POSITIVE	2018
NEGATIVE	2460

Table 6 shows the results obtained from the Sarcasm Labeling process. There several tweets in 2018 are labeled sarcastic, and 2460 others are non-sarcasm. However, it is necessary to balance the dataset by removing tweets with a neutral sentiment or emoji label. The final dataset is described in table 7 below.

Table 7. Final dataset

SENTIMENT LABEL	SARCASM LABEL	QUANTITY
POSITIVE	POSITIVE	103
POSITIVE	NEGATIVE	1097
NEGATIVE	POSITIVE	1094
NEGATIVE	NEGATIVE	106

In table 7, there are tweets with 1200 positive sentiment labels where 103 of them are sarcastic, and 1097 are not. While tweets with a negative sentiment label are 1200 and 1094 were sarcastic, and 106 were not sarcastic.

2.6. Word Embedding Creation

Word embedding is a topic in natural language processing that aims to build the vector representation of word dimensions from various of texts. Word embedding takes on a more expressive and efficient representation by maintaining each word's contextual terms until a low-dimensional vector is obtained. One well-known method, namely Global Vector (GloVE) was proposed by Pennington et al [11].

At the stage of making word embedding, a final dataset of 2400 is used. Each text in the tweet in the dataset will be tokenized and stored in the form of a corpus. The GloVe model that will be built uses the parameters described in Table 8.

Table 8. Parameter of glove model

Emoji label	Quantity
WINDOW	5
OUTPUT DIMENSION	100
LEARNING RATE	0.05
Epoch	30
CORPUS LENGTH	2400

After creating the GloVe model, a document containing a unique word with a 100-vector number vector is generated. This vector document is then used for embedding layers on the CNN architecture.

2.7. Sarcasm Detection Engine

The development stage of the Sarcasm Detection Engine is the last stage of this research. Sarcasm Detection Engine has two main components, namely text sentiment classifier using CNN and Emoji sentiment classifier.

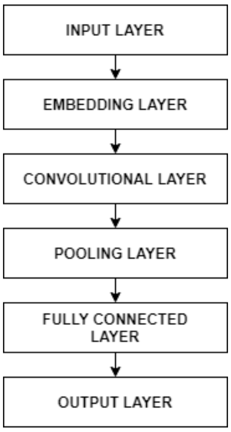


Figure 3. CNN architecture sarcasm detection engine

A researcher first developed the convolutional neural network from NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan, Kunihiko Fukushima, under the name NeoCognitron [9]. The concept of CNN was refined by a researcher from AT&T Bell Laboratories in Holmdel, New Jersey, USA, Yann LeChun, with a CNN model named LeNet that was used by LeChun to detect numbers and handwriting. [12].

CNN is one of the methods in applied deep learning. Like neural networks in general, this system will also be trained with backpropagation. The CNN method has many layers, namely convolution layer, subsampling/pooling layer, and fully-connected layer. CNN also has several activation functions, for example, ReLu and sigmoid functions.

Figure 3 is the CNN architecture that will be used. n this research, we did not use the reference parameters for the existing researches. We have done several experiments, including changing the form of CNN architecture and its respective layers from several experiments, we took the best results, but these results are not the most optimal because this research have not covered all the parameters yet.

Detailed parameters for each layer are explained in Table 9, 10, 11, 12, 13, 14 and 15.

Table 9. Input layer parameter

PARAMETER	VALUE
PADDING ZEROS TO THE LONGEST ELEMENT	TRUE

Table 10. Embedding layer parameter

PARAMETER	VALUE
VOCAB SIZE	2400
INPUT DIMENSION	100
WEIGHT	GLOVE EMBEDDING MATRIX
INPUT LENGTH	35
TRAINABLE	TRUE

Table 11. Convolution 1D layer parameter

PARAMETER	VALUE
NUMBER OF FILTERS	32
KERNEL SIZE	5
ACTIVATION FUNCTION	RELU

Table 12. Pooling 1D layer parameter

PARAMETER	VALUE
POOL SIZE	2

Table 13. Fully-connected layer parameter

PARAMETER	VALUE
NUMBER OF NODES	10
ACTIVATION FUNCTION	RELU

Table 14. Output layer parameter

PARAMETER	VALUE
NUMBER OF NODES	1
ACTIVATION FUNCTION	SIGMOID

Table 15. General CNN architecture parameter

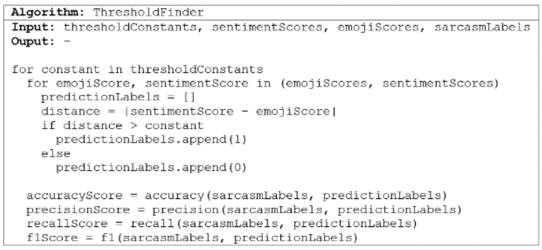
PARAMETER	VALUE
OPTIMIZER	ADAM
LOSS FUNCTION	BINARY CROSS-ENTROPY

Input on the text sentiment classifier is a text feature of training and testing datasets. The training dataset is used to train CNN, and the testing dataset is used to get a sentiment score.

$$emoji\_polarity_t = \frac{\sum pos\_pol\_val_t - \sum neg\_pol\_val_t}{total\_emoji_t} \qquad (6)$$

In the emoji polarity score calculation, each emoji in a tweet will be calculated for its polarity score using equation 6.

The Input on the emoji sentiment classifier is an emoji feature from the testing dataset to calculate the testing dataset's polarity score. Polarity score is the sum of the positive emoji polarity score while it is the sum of the negative emoji polarity values. The function produces a range of values between [-1,1], then it needs to be normalized by using MinMax normalization and resulting values with ranges between [0,1].



```
Algorithm: ThresholdFinder
Input: thresholdConstants, sentimentScores, emojiScores, sarcasmLabels
Ouput: -

for constant in thresholdConstants
  for emojiScore, sentimentScore in (emojiScores, sentimentScores)
    predictionLabels = []
    distance = |sentimentScore - emojiScore|
    if distance > constant
      predictionLabels.append(1)
    else
      predictionLabels.append(0)

  accuracyScore = accuracy(sarcasmLabels, predictionLabels)
  precisionScore = precision(sarcasmLabels, predictionLabels)
  recallScore = recall(sarcasmLabels, predictionLabels)
  f1Score = f1(sarcasmLabels, predictionLabels)
```

Figure 4. Threshold finder pseudocode algorithm

After getting the polarity value of each feature from the testing dataset, the classification of sarcasm is performed by calculating the difference in distance between the text and emoji features. Determination of the optimal distance limit for sarcasm labeling is obtained from the highest f1-score from the interval value obtained in the pseudocode of figure 4.

3. Result and Analysis

To get optimal results from the Sarcasm detection engine model, we conducted several experiments of a sarcasm detection engine component.

The first trial we did was to maximize the hyperparameter value on the CNN model. This experiment uses the architecture mentioned in the proposed method section. This trial was conducted by cross-validation. We are dividing the training data into eo parts, namely training data of 1800 tweets and validation data of 200.

The experiment aims to find an optimal CNN model where the model will not underfitting or overfitting. Some experiments conducted with test data of 400 tweets then obtained the highest accuracy value of 87.5%.

The second trial by comparing the word embedding model. In this study, the model we proposed uses GloVe word embedding, but we also experiment using Word2Vec CBOW word embedding. This experiment aims to find out the optimal word embedding model to be used in the CNN layer embedding. The first trial parameter using output dimensions of 100 and 300. The second trial parameter used additional training data from 379,557 documents in Indonesian Wikipedia. The final test parameter is that the embedding layer's value can be trained or not during the CNN model training phase.

Table 16. Accuracy of CNN using different parameter on word embedding

ALGORITHM	CORPUS	DIMENSION	ACCURACY
TEST – 1 (TRAINABLE EMBEDDING LAYER)			
GLOVE	TWEET	100	87.5%
	TWEET	300	86.2%
	TWEET + WIKI	100	82.9%
	TWEET + WIKI	300	81.2%
WORD2VEC	TWEET	100	85.0%
	TWEET	300	84.3%
	TWEET + WIKI	100	68.0%
	TWEET + WIKI	300	63.0%
TEST – 2 (TRAINABLE EMBEDDING LAYER)			
GLOVE	TWEET	100	86.7%
	TWEET	300	86.0%
	TWEET + WIKI	100	83.0%
	TWEET + WIKI	300	82.1%
WORD2VEC	TWEET	100	81.0%
	TWEET	300	84.5%
	TWEET + WIKI	100	58.7%
	TWEET + WIKI	300	68.0%
TEST – 3 (NON-TRAINABLE EMBEDDING LAYER)			
GLOVE	TWEET	100	50.9%
	TWEET	300	51.7%
	TWEET + WIKI	100	68.7%
	TWEET + WIKI	300	71.7%
WORD2VEC	TWEET	100	60.0%
	TWEET	300	59.2%
	TWEET + WIKI	100	62.5%
	TWEET + WIKI	300	67.7%
TEST – 4 (NON-TRAINABLE EMBEDDING LAYER)			
GLOVE	TWEET	100	58.7%
	TWEET	300	53.2%
	TWEET + WIKI	100	66.0%
	TWEET + WIKI	300	66.5%
WORD2VEC	TWEET	100	59.2%
	TWEET	300	59.2%
	TWEET + WIKI	100	64.5%
	TWEET + WIKI	300	67.7%

From the results of experiments 1 and 2 in table 16, it can be concluded that the number of output dimensions between 100 and 300 shows no difference. The gloVe is superior to Word2Vec but not very significant.

The use of an additional Wikipedia training dataset in experiments 1 and 2 reduces the accuracy of the bulit CNN model. For experiments 3 and 4 in Table 15 shows when using pre-trained embedding layers that use additional Wikipedia training data, it can increase the accuracy of the CNN model built when the embedding layer cannot be trained during the training phase of the CNN model.

From the two experiments conducted, we chose the CNN model using GloVe word embedding, which was trained with only tweet datasets with an output vector length of 100. Obtained an optimal accuracy score of 87.5% for the CNN model architecture that was built.

The selection of the most optimal threshold value for the sarcasm detection engine is made by finding the highest f1-score value for each entered interval value. In this experiment, the increased interval value is set to 0.01 in the range [0,1].

Table 17. Threshold F1-Score, precision, recall value

THRESHOLD VALUE	F1-SCORE	PRECISION	RECALL
0.37	87.59%	90.58%	84.80%
0.38	87.59%	90.58%	84.80%
0.26	87.47%	87.68%	87.25%
0.28	87.41%	88.06%	86.76%

Table 17 shows the results of the four threshold values with the highest f1-score value. The best detection engine treshold values sarcasm range from 0.37 to 0.38 with an f1-score of 87.59%, a precision of 90.58%, recall of 84.80%, respectively.

Table 18. Expert validation

TEXT	SYSTEM	GROUND TRUTH
JOKOWI KAHAN GAYA 🤣🤣. NIH URUSIN DULU: 1. KEMISKINAN 2. PENGANGGURAN 3. KORUPSI 4. DEMOKRASI	1	1
GK PAPA BRO YG NYALONIN PARA ORG SAKIT HATI UDAH MABOK JABATAN PARTAI GUREM LAGI 😂😂😂	1	1
YANG DISALAHIN PAK JOKOWI 😬😬😬	0	0
KERENNNNNN LANJUTKAN BAPAK JADIKAN INDONESIASEMAKIN MAJU😘😇	0	0

To validate the model built, we answered all of the tweets that were approved by the model for three expert approval. Table 18 is a sample tweet that was tested by an expert and from the sarcasm label system. The sample shows the result of the labeling of sarcasm by the system and the expert's judgment, which is used as the ground truth, where label 1 indicates that emojis in the tweet match the sentiment label. Based on the sample, the proposed system has worked well. This is indicated by the similarity of the system label with the ground truth label.

4. Conclusion

This research has made a sarcasm detection engine for Indonesian tweets with the motivation to detect sarcasm based on textual and emoji features. We proposed a supervised machine learning approach using the Convolutional Neural Network to calculate the value of sentiment polarity and emoji weighting to calculate the emojis polarity score. Furthermore, the method we propose focuses on textual features and emojis for finding sarcastic tweets. We also conducted experiments on the parts of the detection engine sarcasm, namely the Convolutional Neural Network.

The Convolutional Neural Network architecture that we built consists of an embedding layer using GloVe with a vector length of 100 and has been trained using tweets dataset. The accuracy of the Convolutional Neural Network model built was 87.5%. The accuration shows that the model of the Convolutional Neural Network that was build can determine the value of sentiment polarity very well.

The sarcasm detection engine that we have built has an f1-score of 87.59%. Thus sarcasm detection engine that we built in this research has a good level of accuracy. This is proven by validating the expert directly and having results that match the expert's judgment.

From research conducted that with the textual and emoji features, we can determine whether an expression is a sarcasm or not.

In our research, we realized the model that was built was not perfect. Therefore, it is necessary to do further research on the sarcasm detection engine that has been built. In the future, we can integrate the engine that we have built with sarcasm detection based on textual features only, where a word in a tweet has a different polarity value far from its closest neighbor. It can be categorized as an expression of sarcasm. This needs to be done so that the results of the engine will be more accurate. Expert linguists should annotate the dataset so the dataset is more valid annotated.

References

[1] S.K. Bharti, R.Naidu & K.S Babu,"Hyperbolic Feature-based Sarcasm Detection in Tweets: A Machine Learning Approach", In Proceeding of the 2017 14th IEEE INDICON, 2017.

[2] Liu, B. Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies, 5(1), 1–167. 2012.

[3] Feldman, R. (2013). Techniques and applications for sentiment analysis. Communications of the ACM, 56(4), 82–89.

[4] Saha, S. et al.“Proposed Approach for Sarcasm Detection in Twitter’, Indian Journal of Science and Technology, 10(25), DOI: 10.17485/ijst/2017/v10i25/114443,. July 2017.

[5] D.A.P. Rahayu, S. Kuntur & N. Hayatin, "Sarcasm Detection on IndonesiaTwitter Feeds", In Proceeding of EECSI 2018, 2018.

[6] S.K. Bharti, B. Vachha, R.K. Pradhan, K.S Babu & S.K. Jena, “Sarcastic sentiment detection in tweets streamed in real time: a big data approach”, Digital Communications and Networks, vol. 2, issue 3, pp. 108-121. 2016.

[7] W. Medhat, A. Hassan & H. Korashy, “Sentiment analysis algorithms and applications:A survey”, Ain Shams Engineering Jornal, vol. 5, issue 4, pp. 1093-1113. 2014.

[8] H.M.K. Kumar & S. Harish, “Sarcasm classification: A novel approach by using Content Based Feature Selection Method”, Procedia Computer Science, vol. 143, pp. 378-386, 2018.

[9] M.Y. Manohar & P. Kulkarni, “Improvement sarcasm analysis using NLP and corpus based approach”, In proceeding of 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 2017.

[10] Dalyani.G.Geeta, “Emoticon-based unsupervised sentiment classifier for polarity analysis in tweets”, International Journal of Engineering Research and General Science Volume 2, Issue 6, October-November, 2014.

[11] M.Naili, A.H Chaibi, H.H.B. Ghezala. “Comparative Study of Word Embedding Methods in Topic Segmentation”, In International Conference on KES 2017, pp. 340-349, 2017.

[12] Suartika, I. W., Wijaya, A. Y. and Soelaiman, R. (2016) “Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101”, Jurnal Teknik ITS, 5(1), pp. 65–69.DOI: 10.12962/j23373539.v5i1.15696.

[13] L. Gatti, M. Guerini, and M. Turchi, “Sentiwords: Deriving a high precision and high coverage lexicon for sentiment analysis,” IEEE Transactions on Affective Computing, vol. 7, no. 4, pp. 409–421, Oct 2016.

[14] Novak, P. K. et al. “Sentiment of emojis”, PLoS ONE, 10(12), pp. 1–22. DOI: 10.1371/journal.pone.0144296. 2015.