Comparison of Accuracy between Convolutional Neural Networks and Naïve Bayes Classifiers in Sentiment Analysis on Twitter

P.O. Abas Sunarya[1], Rina Refianti[2], Achmad Benny Mutiara[3], Wiranti Octaviani[4]
Dept. of Informatics Engineering, STMIK Raharja Jl. Jenderal Sudirman No. 40, Tangerang 15117, Indonesia[1]
Faculty of Computer Science and Information Technology
Gunadarma University, Jl. Margonda Raya No. 100, Depok 16424, Indonesia[2, 3, 4]

***Abstract*—The needs and demands of the community for the ease of accessing information encourage the increasing use of social media tools such as Twitter to share, deliver and search for information needed. The number of large tweets shared by Twitter users every second, making the collection of tweets can be processed into useful information using sentiment analysis. The need for a large number of tweets to produce information encourages the need for a classifier model that can perform the analysis process quickly and provide accurate results. One algorithm that is currently popular and is widely used today to build classifier models is Deep Learning. Sentiment analysis in this research was conducted on English-language tweets on the topic "Turkey Crisis 2018" by using one of the Deep Learning algorithms, Convolutional Neural Network (CNN). The resulting of CNN classifier model will then be compared with the Naïve Bayes Classifier (NBC) classifier model to find out which classifier model can provide better accuracy in sentiment analysis. The research methods that will be carried out in this research are data retrieval, pre-processing, model design and training, model testing and visualization. The results obtained from this research indicate that the CNN classifier model produces an accuracy of 0.88 or 88% while the NBC classifier model produces an accuracy of 0.78 or 78% in the testing phase of the data test. Based on these results it can be concluded that the classifier model with Deep Learning algorithm produces better accuracy in sentiment analysis compared to the Naïve Bayes classifier model.**

***Keywords*—Sentiment-analysis; convolutional neural network; deep learning; Naïve Bayes classifier**

I. INTRODUCTION

The needs and demands of the community for the ease of accessing information encourage the increasing use of social media facilities to share, deliver, and search for information needed. One of the popular social media that is widely used by people from various backgrounds is Twitter. Twitter provides facilities with features that are easy to understand for users to publish daily activities, inform a news or fact, and express opinions. This makes Twitter still popular today.

Twitter receives tweets from users' as many as 55 million messages every day [1]. The number of large tweets shared by Twitter users every second, making a collection of tweets can be processed into useful information such as to find out a review or public opinion about a particular product, service, or topic.

The process of processing tweet data to get information requires a method that can find patterns of linkages and classify these tweets, one of which uses sentiment analysis. Sentiment analysis is done to classify data into positive, negative and neutral classes.

The need for a large number of tweets to produce information encourages the need for a classifier model that can perform the analysis process quickly and provide accurate results. One algorithm that is currently popular and widely used today to build classifier models is Deep Learning. Deep Learning Algorithm, one of them is Convolutional Neural Network (CNN) which utilizes the Neural Network concept to carry out many learning processes applied in analyzing and predicting processes. The CNN algorithm is inspired by the workings of human brain neurons which consist of several layers. Each neuron is interconnected and will forward information between layers. Information will go through the iteration and distribution process to each subsequent layer to produce the final output as needed. This iteration process helps the machine to learn and identify information so that it will produce a classifier model that can do the classification process of new data with a good level of accuracy.

The CNN algorithm is generally more implemented to analyze and predict two-dimensional objects (images) but there are several studies that apply the CNN algorithm to one-dimensional objects such as text. One example of research that applies the CNN algorithm in text classification is the research of Yoom Kim (2014) [2]. Based on the research, it was found that the classifier model with CNN algorithm showed good classification performance in text classification (such as sentiment analysis) and since it became the basic standard in text classification.

Based on the above background, in this research will use Convolutional Neural Network (CNN) and Naïve Bayes Classifier (NBC) algorithms in the sentiment analysis process using Twitter data which is expected to produce classifier models with good accuracy. Accuracy results from the CNN classifier model will then be compared with the results of the accuracy of the NBC classifier model; so that it can be seen which algorithm is capable of producing classifier models with better accuracy values.

The limitations of the problem in this research can be formulated as follows:

1) The sentiment analysis process was carried out related to the topic "Turkey Crisis 2018" with a tweet obtained from Twitter totalling 45,443 data based on a hash tag (#) relating to the topic taken.

2) The tweet used in this research was only an English tweet.

3) The process of sentiment analysis and the making of the classifier model in this research use the Python programming language version 3.6.

4) Classification of tweet data obtained into positive, negative, and neutral classes using the Text Blob library in Python.

5) Using Deep Learning algorithm, Convolutional Neural Network (CNN) and Machine Learning algorithm, Naïve Bayes Classifier (NBC) to build classifier models that can classify sentiments of new data.

6) Compare the results of the accuracy values produced by the CNN classifier model with the results of the accuracy of the NBC classifier model.

7) Visualize the comparison results of the accuracy from CNN and NBC models into tables and graphs.

The aim of this research is to use Deep Learning algorithm, namely Convolutional Neural Network (CNN) in the sentiment analysis process on English tweets related to the topic "Turkey Crisis 2018" on Twitter data and compare the results of the accuracy values obtained from the CNN classifier model with the results of accuracy values from the Naïve Bayes Classifier model to find out which classifier models produce better accuracy values in text classification.

In the rest of paper, we show briefly the literature review and related work in Section II. In Section III the research methodology is presented. The implementation and results related to our research are also shown in Section IV. The last section is conlusion and future work of our research.

II. Literature Review

A. Sentiment Analysis

According to B. Liu (2010) [3], sentiment analysis or opinion mining is a process of understanding, extracting and processing textual data automatically to get information on sentiments contained in an opinion sentence. Sentiment analysis is done to see opinions or trends of opinion on a problem or object by someone, whether they tend to have a negative or positive opinion or opinion.

As in [4], the basic task in sentiment analysis is to classify the polarity of the text in documents, sentences, or features, namely whether the opinions expressed in the document, sentence or feature are positive, negative or neutral.

B. Twitter

Twitter is a website that is a service from microblog, which is a form of blog that limits the size of each post, which provides facilities for users to be able to write messages in Twitter updates containing only 140 characters. Twitter was founded by three people, namely Jack Dorsey, Biz Stone, and Evan William in March 2006 and was launched in July of the same year.

All users can send and receive tweets via Twitter sites, compatible external applications (cell phones), or with short messages (SMS) available in certain countries. Users can write messages by topic using the # (hashtag). Whereas to mention or reply to messages from other users can use the @ (et) sign.

The characteristics of a microblogging or Twitter, which has a status update commonly referred to as tweet totaling 140 characters shorter than other media; can comment on the tweet made by following by using reply, then it can be written using the RT @ username function; have their own way of sharing photos and videos commonly referred to as tweetpic as in [5].

C. Naive Bayes Classifier

Naïve Bayes Classifier (NBC) is a text mining method that can be used to solve opinion mining problems. NBC can be used to classify opinions into positive and negative opinions. NBC can function properly as a method of text classifiers.

The Naïve Bayes classification algorithm utilizes the probability theory proposed by British scientist Thomas Bayes, which predicts future probabilities based on past experience. The simple NBC algorithm and its high speed in the training and classification process make this algorithm interesting to use as a classification method. The classification process is usually divided into two phases, namely, learning and test. In the learning phase, some of the data that has been known for the data class is fed to form an approximate model. Then in the test phase the model that has been formed is tested with some other data to determine the accuracy of the model.

In the Naïve Bayes Classiffier algorithm each tweet is represented by a pair of attributes "x_1, x_2, x_3, ... x_n" where x_1 is the first word, x_2 is the second word and so on. Whereas V is a set of tweet categories. At the time of classification the algorithm will look for the highest probability of all categories of tweets tested ($\boldsymbol{v}_{\mathrm{MAP}}$), where the equation is as follows:

$$v_{MAP} = arg\max_{v_j \in V} \frac{P(x_1,x_2,x_3 \cdots, x_n|v_j)P(v_j)}{P(x_1,x_2,x_3 \cdots, x_n)} \quad (1)$$

For $P(x_1,x_2,x_3,.., x_n)$ the value is constant for all categories (v_j) so that the equation can be written as follows:

$$v_{MAP} = arg\max_{v_j \in V} P(x_1, x_2, x_3 \cdots, x_n|v_j)P(v_j) \quad (2)$$

The above equation can be simplified as follows:

$$v_{MAP} = arg\max_{v_j \in V} \prod_{i=1}^{n} P(x_i|v_j)P(v_j) \quad (3)$$

where v_j = category of tweets j = 1, 2, 3, ... n, in this study j = 1 indicates a category of negative sentiment tweets, j = 2 indicates a category of positive sentiment tweets and j = 3 indicates a category of neutral sentiment tweets:

$P(x_i \mid v_j)$ = probability xi in the category v_j ;

$P(v_j)$ = probability of v_j .

For $P(v_j)$ and $P(x_i \mid v_j)$ it is calculated during the training where the equation is as follows:

$$P(v_j) = \frac{|docs_j|}{|example|} \quad (4)$$

$$P(x_i|v_j) = \frac{n_k+1}{n+|vocabulary|} \quad (5)$$

where

$P(v_j)$ = The probability of each document against a set of documents;

$P(x_i \mid v_j)$ = The probability of the occurrence of the word x_i in a document with the class category v_j ;

| docs | = number of documents in each category j ;

|example| = number of documents from all categories;

n_k = number of times the frequency of occurrence of each word;

n = number of frequency of occurrence of words from each category

There are several forms of representation of the Naïve Bayes Classifier method, including:

1) Gaussian naive bayes: Gaussian Bayes are usually used to represent the conditional probability of the continue feature in a class (x_i | y), and are characterized by two parameters: mean and variant.

2) Bernaulli naïve bayes: In Naïve Bayes Bernaulli, weighting is carried out using binaries (0 and 1) in weighting each term, this is different from the calculation of frequency terms that do weighting on each term.

3) Multinomial naïve bayes: Multinomial Naïve Bayes assumes independence between the appearance of words in a document, without taking into account the order of words and context of information in sentences or documents in general. Besides this method takes into account the number of occurrences of words in the document.

The Naive Bayes algorithm that is often used for text mining is Multinomial Naive Bayes. Multinomial Naïve Bayes is one of the specific methods of the Naïve Bayes method. Multinomial Naïve Bayes is also a supervised learning machine in the process of classifying text by using the probability value of a class in a document.

D. Deep Learning

Deep Learning is a branch of science learning based on artificial neural networks (ANN) or it can be said that the development of ANN teaches computers to be able to take actions that are considered natural by humans. For example, to learn from examples. In deep learning, a computer can learn to classify directly from images, sounds, texts or even videos. A computer is trained using data sets labeled and the numbers are very large which can then change the pixel value of an image into an internal representation or feature vector where classification can be used to detect or classify patterns at input input [6][9][10][11].

Deep learning method is a method of learning representation with several levels of representation, where representation forms a neural network architecture field that contains many layers (layers). The deep learning layer consists of three parts, namely the input layer, hidden layer, and output layer. In the hidden layer can be made in layers to find the right algorithm composition to minimize errors in output [6].

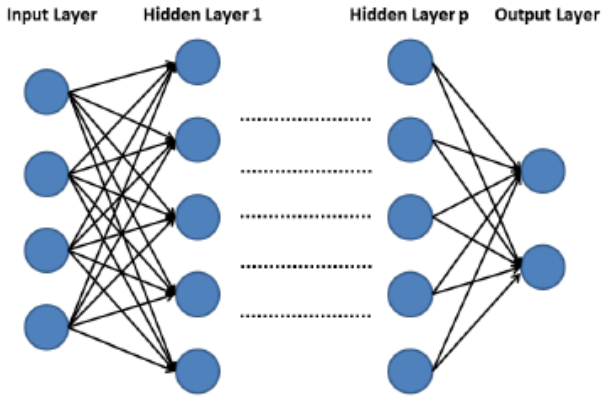

Fig 1. Deep Learning Layers.

Fig. 1 illustrates deep learning layers that have p + 2 layers (p hidden layer, 1 input and 1 output layer). Blue circles represent neurons. There are one or more neurons in each layer. These neurons will be connected directly to other neurons in the next layer [6].

E. Convolutional Neural Network

Convolutional Neural Network (CNN / ConNet) is one of the deep learning algorithms which is the development of the Multilayer Perceptron (MLP) which is designed to do data into two dimensions, for example: images or sound. Convolutional Neural Network is used to classify the labeled data by using supervised learning method, the way it works is that there is training data and there are variables that are targeted so that the purpose of this method is to group data into existing data.

In general, the CNN layer type is divided into two parts, namely:

1) Feature extraction layer (feature extraction layer): The image that is located at the beginning of the architecture is composed of several layers and in each layer arrangement of the neurons connected to the local region (local region) of the previous layer. The first type of layer is the convolutional layer and the second layer is the pooling layer. At each layer the activation function is applied with its intermittent position between the first and second types. This layer accepts image input directly and processes it until it produces a vector output to be processed in the next layer.

2) Classification layer: This layer is composed of several layers which in each layer are composed of fully connected neurons with other layers. This layer receives input from the output of the image feature extraction layer in the form of a vector which is then transformed as in the Multi Neural Network with the addition of several hidden layers. Output results in the form of class accuracy for classification.

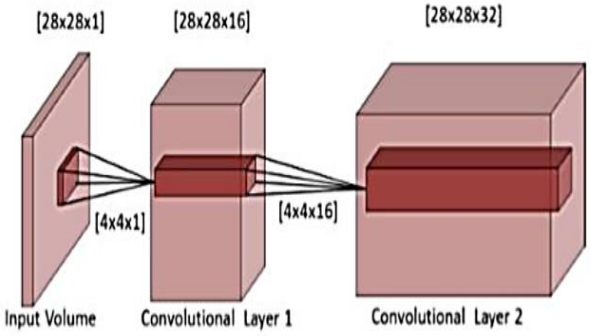

Fig 2. Examples of Convolutional Layer Diagrams [7].

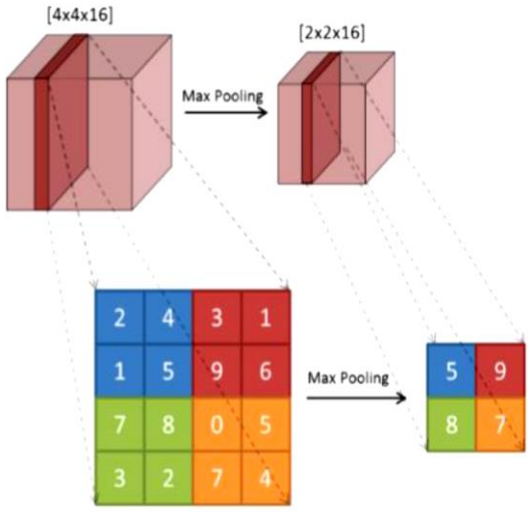

Fig 3. Examples of MAX Pooling Layer Diagrams [7].

As in [7], Convolutional Layer is a main core of CNN, where this layer has a collection of filters that can be used to study input images. Through this layer, the feature will be extracted and then proceed to the next layer in order to extract more complex features. Examples of Convolutional Layer diagrams can be seen in Fig. 2 where the input image size given is 28x28 and a 4x4 filter or kernel.

Pooling Layer is a resizing process that is a process to change the size of different input images, one of them is using the MAX operation. This aims to help reduce the number of parameters and calculation times needed when training the network as well as the work of Bui and Chang in [7]. An example of a Pooling Layer diagram can be seen in Fig. 3.

In Fig. 3, the entered image is 4x4 in size and then resized into 2x2-sized image with a depth of 16. Each value is at Max Pooling, for each 4 pixels a maximum value is taken. Seen in Fig. 3 at 4 pixels in blue, the maximum value to be taken is 5. At 4 pixels in red, the maximum value that will be taken is 9. In pixels in green, the maximum value that will be taken is 8. On pixels in orange, the maximum value to be taken is 7. So as to produce a reduced image.

And the third layer on CNN is Fully Connected Layer, where this layer takes all the neurons in the previous layer (Convolutional Layer and MAX Pooling Layer) and connects them to each single neuron that exists, as we can see in [8].

III. Research Method

The process of designing a classifier model for sentiment analysis in this research consists of five stages:

A. Data Retrieval

The first stage of the process of designing this classifier model is data retrieval using the Twitter API service. The Python programming language has provided tweepy library that can facilitate retrieval of data from Twitter. Data is then saved in .csv or .txt format.

B. Pre-Processing

The second stage is pre-processing, namely the stage where the tweets that have been obtained will be extracted and cleaned from noise, namely random or variant errors in measured variables consisting of RT components, hashtag, digits, user (@), punctuation, url, and others components that are considered to interfere with the tweets classification process. Removing noise object is an important goal of cleaning data because noise inhibits most types of data analysis. The flow of the cleaning process can be seen in Fig. 4 below.

Tweets that have been through the cleaning process will then be classified into three categories of sentiment class namely positive, negative and neutral using the TextBlob library. The flow of the data classification process can be seen in Fig. 5 below.

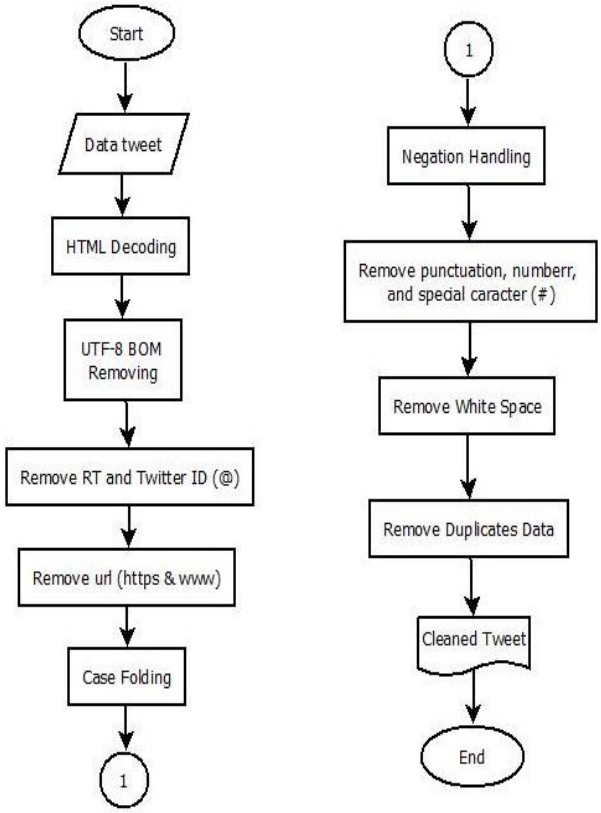

Fig 4. Flow of Data Cleaning.

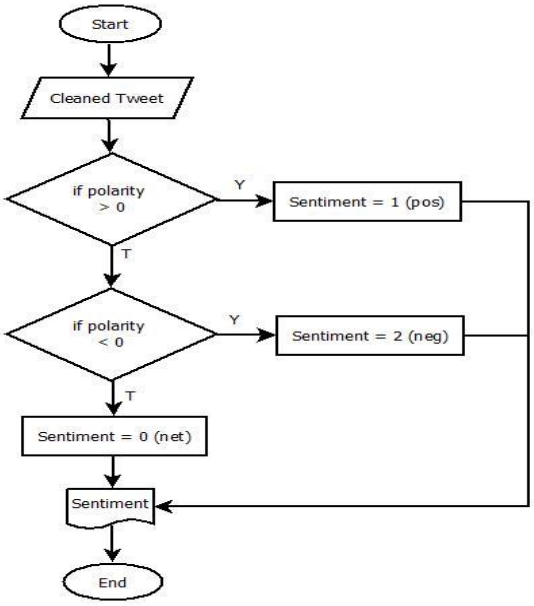

Fig 5. Flow of Data Classification with TextBlob.

C. Model Design and Training

The third stage is the design and training of classifier models, the tweets that have been classified as sentiment classes will be divided into three parts, namely, data train, validation data, and test data. Data train is used to train new classifier models using the Convolutional Neural Network algorithm and the Naïve Bayes Classifier.

D. Testing Model

The fourth stage is the testing phase of the classifier model that has been trained using test data by looking at the value of the accuracy produced. Calculation of accuracy values is done using Confusion Matrix to see how much accuracy is produced by the two classifier models in the training and testing process so that it can be known which model produces better accuracy in sentiment analysis.

E. Visualization

In the final stage, it displays the results in the form of diagrams, graphs and tables.

IV. Implementation and Results

A. Data Retrieval

Retrieving data from Twitter is first done by making a Twitter API connection. The first step that must be done is to create an application on Twitter by visiting the https://apps.twitter.com/ site to get the keys and access tokens used to access the Twitter API.

After getting the key and access token, it takes a Python library that can implement the Twitter API call, one of which is the tweepy library. The next step is to open the Spyder software and install the tweepy library to be able to pull data from Twitter using the Twitter API.

The next step is to retrieve tweets from Twitter based on the hashtag (#) or predefined keywords. Tweets taken are only English-language tweets, taken randomly from ordinary users or Twitter's official media accounts. The topics discussed in this sentiment analysis were "Turkey Crisis 2018" and several hashtags used to search tweets including #TurkeyCrisis, #TurkeyLira, #Turkey, #Erdogan, and #Trump. The tweets that were successfully retrieved were English tweets totalling 45,443 data.

B. Pre-Processing

Data from Twitter that has been taken next will go through the pre-processing stage which consists of the cleaning process of tweets and the classification process of tweets based on positive, negative, or neutral sentiment classes using the TextBlob library. The purpose of pre-processing data is to transform raw data into a format suitable for analysis.

C. Cleaning Data

At this stage the cleaning process of tweet data from noise is carried out, namely random or variant errors in measured variables consisting of RT components, hash tag, digits, user (@), punctuation, url, and other components that are considered to interfere with the tweets classification process.

Tweet data obtained from Twitter often contains components that are not needed and can interfere with the classification process of tweets so that the need for deletion of these components. In the Python programming language, the data cleaning process can use the Beautiful Soup library. After going through the cleaning process, the tweet initially amounted to 45,443 to 33,107 clean tweets.

D. Data Classification using TextBlob

The next stage after cleaning the data tweet is the data classification stage. Tweets that have been cleared from noise components will then be classified to be divided into three sentiment classes, namely positive (1), negative (2) and neutral (0) classes. Data classification at this stage utilizes the TextBlob library. TextBlob classifies tweets into three sentiment classes based on their polarity.

Tweets will be classified into positive sentiment class if the polarity sum of each word in the sentence produces a value greater than 0 it will be labeled 1. The Tweet will be classified into the negative sentiment class if the polarity sum of each word in the sentence produces a value less than 0 it will be labeled 2. Tweets will be classified into neutral sentiment class if the polarity sum of each word in the sentence produces a value equal to 0 it will be labeled 0.

From the 33,107 tweeted and classified tweets, a neutral category of 14,443 tweets, a positive category of 12,142 tweets, and a negative category of 6525 tweets were obtained. The tweet data that has been classified will be equalized for each class of sentiment because the tweet data is uneven and tends to be neutral. Alignment of the number of tweets will follow the amount of data in the sentiment class with the least data, namely the negative class with the number of data 6525. After leveling, the number of tweets for each sentiment class is 6525.

E. Model Design and Training

The tweet data that has been through the cleaning process and the classification process using the TextBlob library will then be used to build a classifier model using the Convolutional Neural Network (CNN) algorithm and the Naïve Bayes Classifier (NBC) algorithm.

F. Split Dataset

Tweets that have been classified as sentiment class will be divided into training data, validation data, and test data which will later be used in designing classifier models using the CNN and NBC algorithms.

The data split in this research was done using the Python library, the Scikit-learn library with the split_train_test method. Data will be divided into three parts including:

1) Data Train: the data set used for the learning process by the classifier model.

2) Data Validation: the data set used to set the parameters of the classifier and provide an unbiased evaluation of a model.

3) Data Test: the data set used to assess the performance of the final model.

G. Designing the CNN Classifier Model

The process of constructing a classifier model for analysis sentiments using the Convolutional Neural Network algorithm consists of several stages, namely importing datasets, dividing datasets, feature extractions using word2vec, tokenization and padding sequences, designing layers in models, model training and evaluation, model testing and visualization. Fig. 6 shows the flow of the design of the classifier model with the CNN algorithm.

H. CNN Model Training and Evaluation

The training phase is carried out as a process to find the patterns of linkages between input variables and output variables from the data studied so that later this model can be used to analyze sentiment on new data. Based on the data splitting at the beginning, the data train amounted to 15,655 data with a 33.05% negative share, 33.45% positive, 33.50% neutral. The training process will be carried out with 10 epochs and the results of the training model will be stored whenever an increase in the accuracy value is generated at each epoch.

From the eight (8) classifier models generated from the training and validation process, the best model is the model produced at the 3rd epoch because it produces the best accuracy value of 0.89 and a loss value of 0.33 before the classifier model become overfitting.

Table I will display the accuracy value and loss value generated by the CNN classifier model at each epoch during the validation process.

The best classifier model that is produced, namely the model in the 3rd epoch will then be used to test the test data.

I. CNN Model Trial of Data Test

The CNN Classifier model that has been trained and evaluated in the previous stage will be tested with test data to see whether the resulting accuracy value will be as good as the accuracy value at the training and validation stages. The test data used in this research is the data obtained based on the split process in the initial part consisting of 1957 data with a 34.18% negative, 33.67% positive, and 33.50% neutral. Calculation of accuracy values for the test data is done using Confusion Matrix to determine the value of precision, recall and f1-score generated by the model.

From the result of testing using test data, the CNN classifier model give an accuracy value of 0.88 or 88% with a loss value of 0.33 or 33%. Fig. 7 will display the classification report from the CNN classifier model test in the test data using the Confusion Matrix calculation.

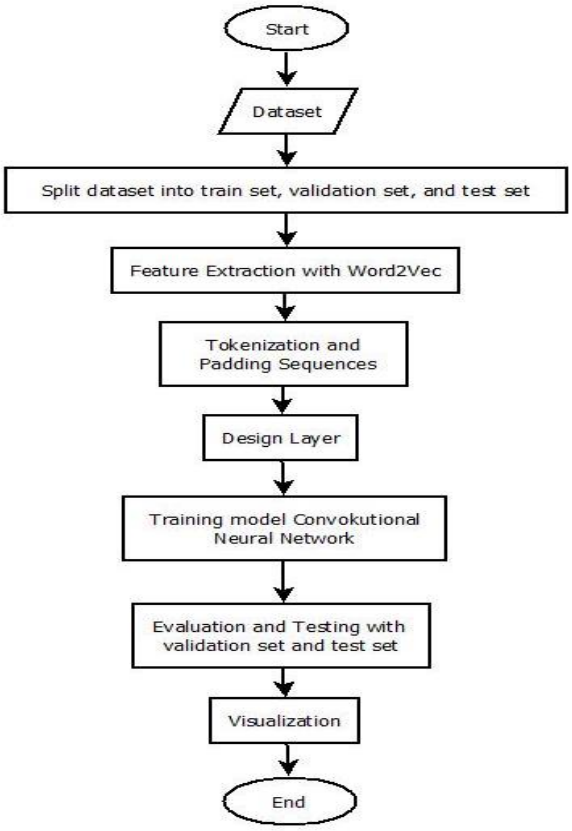

Fig 6. Design Flow of CNN Model Classifier.

TABLE I. ACCURACY AND LOSS VALUE OF DATA VALIDATION IN EACH EPOCH

Epoch	Validation Accuracy	Validation Loss
1	0.785	0.569
2	0.869	0.390
3	0.893	0.332
4	0.907	0.370
5	0.893	0.424
6	0.916	0.429
7	0.923	0.465
8	0.926	0.495
9	0.929	0.502
10	0.929	0.525

Confusion Matrix

	predicted_netral	predicted_positive	predicted_negative
netral	558	34	37
positive	22	572	65
negative	40	36	593

Classification Report

	precision	recall	f1-score	support
netral	0.90	0.89	0.89	629
positive	0.89	0.87	0.88	659
negative	0.85	0.89	0.87	669
avg / total	0.88	0.88	0.88	1957

Fig 7. CNN Model Classification Report For Data Test.

J. Visualization of CNN Model Accuracy Results

Visualization is made to make it easier to understand the results obtained from the training process to the trial. Fig. 8 shows a comparison graph of the validation accuracy value of the train accuracy at each epoch.

Based on Fig. 8, it can be concluded that the accuracy value produced by the classifier model during the training process increases at each epoch. While the accuracy value generated during the validation process has increased in the first to fourth epoch, but at the 5th epoch accuracy has decreased. At the 6th epoch, the accuracy value increases again and starts from the 7th to 10th epoch, the accuracy value is stable.

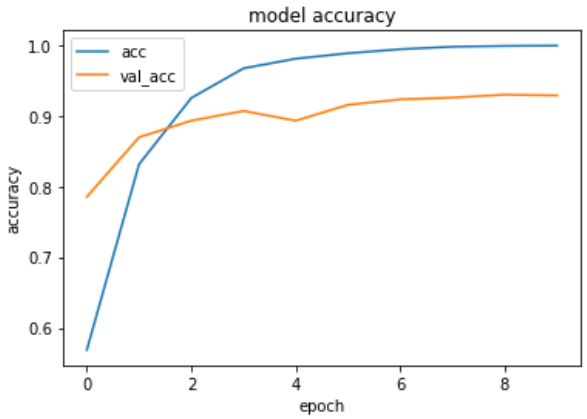

Fig 8. Comparison Graph of Validation Accuracy against Train Accuracy.

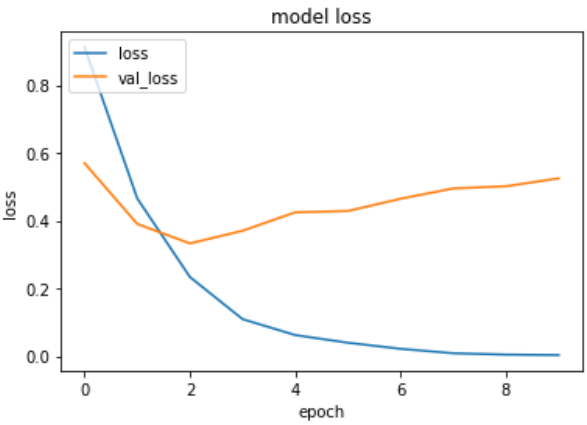

Fig 9. Comparison Graph of Validation Loss against Train Loss.

Based on Fig. 9 it can be concluded that the loss value generated during the training process has decreased in each epoch while the loss value generated during the validation process has decreased to the 3rd epoch but starting from the 4th to 10th epoch loss values are increasingly experiencing which increase indicates that the classifier model is overfitting. The best model is the model produced at the 3rd epoch because it produces the best accuracy value of 0.89 and a loss value of 0.33 before the classifier model become overfitting.

K. Design of NBC Classifier Model

The process of building a classifier model with the Naïve Bayes Classifier algorithm in this research consisted of several stages, namely importing datasets, dividing datasets, feature extractions, conducting model training on several gram n-features, testing the accuracy of each gram n-feature model, validating, testing model and visualization. In Fig. 10 is shown the process flow design of the Naïve Bayes classifier model.

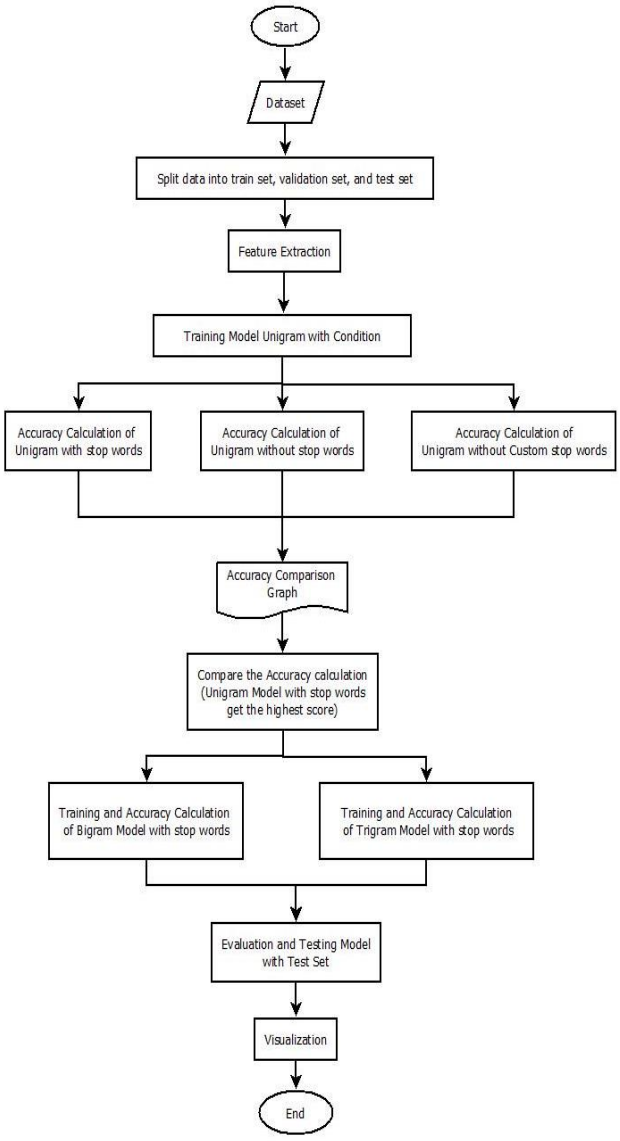

Fig 10. Design Flow of the Naïve Bayes Classifier Model.

	negative	positive	netral	total
turkey	4717	4566	4835	14118
the	3321	3562	2848	9731
to	2452	2441	2037	6930
in	1973	1765	1739	5477
of	1653	1884	1519	5056
and	1764	1791	1478	5033
is	1788	1778	1252	4818
you	1113	976	789	2878
on	1128	891	842	2861
not	1217	878	753	2848

Fig 11. Custom Stop Words List.

L. NBC Model Training and Evaluation

The training phase is carried out on several N-gram models to get the model with the best accuracy value which will then be evaluated with the Naïve Bayes Classifier algorithm. N-gram is a method for retrieving bits of letter characters of n from a word. N-gram has three types of processing models in a sentence; the type of processing includes Unigram for separating one word in a sentence, Bigram for separating two words in a sentence, and Trigram for separating three words in a sentence.

Classifier training models will be carried out on the N-Gram model with several conditions, among others, the unigram with stop words, unigram without stop words, and unigram without custom stop words. Custom stop words are stop words derived from 10 words that most often appear on the corpus. In Fig. 11, it shows the custom stop words list in this study.

Based on the training and validation process carried out on the three unigram models with the conditions mentioned, namely with stop words, without stop words, and without custom stop words, the highest accuracy was generated by the unigram with stop words model with an accuracy value of 77.82% with the number feature 3000.

After getting the results that the highest accuracy value is generated from the unigram with stop words model, then an experiment will be conducted to conduct training and accuracy testing on the bigram and trigram with stop words models to see whether there will be an increase in accuracy.

Based on the results of the accuracy obtained from the unigram, bigram, and trigram with stop words training and validation processes, the best accuracy values for each n-gram model are as follows:

1) Unigram: on 3,000 features with validation accuracy of 77.82%

2) Bigram: on 5,000 features with validation accuracy of 75.27%

3) Trigram: on 5,000 features with validation accuracy 74.71%

The Unigram with stop words classifier model that produces the best accuracy values will then be used to test on the test data.

M. NBC Model Trial of Data Test

Based on the training process and the validation of the NBC classifier model in the previous stage, it was known that the unigram with stop words model produced the highest accuracy in 3000 features. At this stage, the accuracy of the classifier model will be tested for the test data. The test data used in this research is the data obtained based on the split process in the initial part consisting of 1957 data with a 34.18% negative, 33.67% positive, and 33.50% neutral. Accuracy testing of the data test was done using Confusion Matrix to determine the precision, recall and f1-score values produced by each model.

The unigram NBC classifier model which was tested with the data test gave an accuracy value of 0.78 or 78%. Fig. 12 will display the classification report from the NBC classifier model test in the test data using the Confusion Matrix calculation.

N. Visualization of NBC Model Accuracy Results

Fig. 14 shows a comparison graph of the accuracy results obtained from the training and validation process carried out on the three unigram models with the conditions mentioned, namely with stop words, without stop words, and without custom stop words.

Based on Fig. 13 can be concluded as follows:

1) The best accuracy of unigram without stop words is the 13,000 feature with an accuracy value of 74.55%

2) The best accuracy is unigram with stop words, namely the 3000th feature with an accuracy value of 77.82%

3) The best unigram accuracy without custom stop words is the 3000th feature with an accuracy value of 77.72%

```
null accuracy: 66.79%
accuracy score: 77.82%
model is 11.04% more accurate than null accuracy
---------------------------------------------------------------------
Confusion Matrix

          predicted_negative  predicted_positive  predicted_netral
negative                 557                  72                51
positive                  70                 520                37
netral                   120                  84               446
---------------------------------------------------------------------
Classification Report

              precision    recall  f1-score   support

    negative       0.84      0.69      0.75       650
    positive       0.77      0.83      0.80       627
      netral       0.75      0.82      0.78       680

avg / total       0.78      0.78      0.78      1957
```

Fig 12. Unigram Classification Report in 3000 Feature against Data Test.

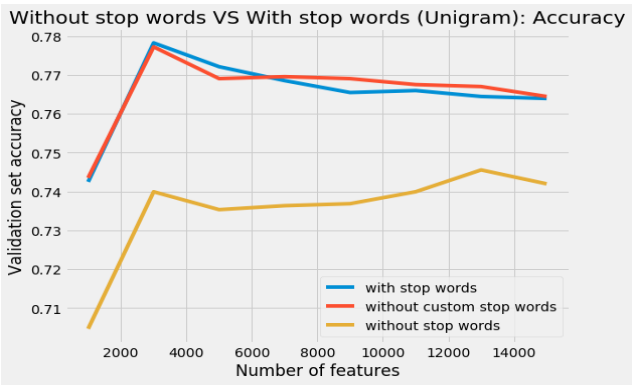

Fig 13. Comparison of Unigram Model Accuracy with Conditions.

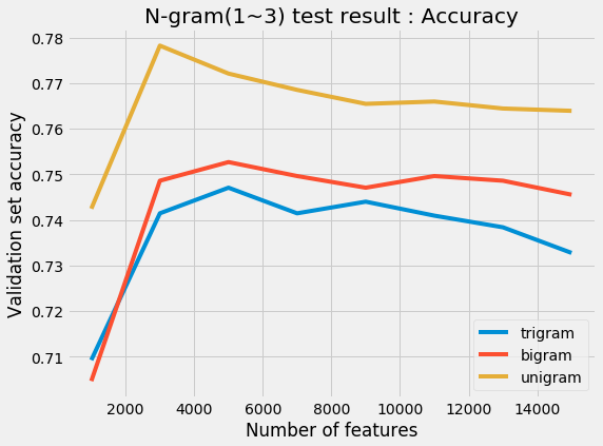

Fig 14. Comparison of Model Accuracy of Unigram, Bigram, and Trigram.

After getting the results that the highest accuracy value is generated from the unigram with stop words model, then an experiment is conducted to test the accuracy of the bigram and trigram with stop words models to see whether there will be an increase in accuracy. In Fig. 14 can be seen the comparison of the results of accuracy produced by unigram, bigram, and trigram with stop words models.

Based on Fig. 14 can be concluded as follows:

1) The best accuracy of Unigram on the 3000 feature with an accuracy value of 77.82 %.

2) Best accuracy of Bigram on the 5000 feature with an accuracy value of 75.27 %.

3) The best accuracy of Trigram on the 5000 feature with an accuracy value of 74.71%.

4) From the three classifier models, the unigram with stop words model produces the best accuracy values.

O. Comparison of CNN and NBC Classifier Model Accuracy Values in Data Test

The final result of this research is to find out which classifier model produces better accuracy in sentiment analysis. Based on the classification report, the accuracy testing of the test data carried out in the previous stage shows that the CNN classifier model produces an accuracy value of 0.88 or 88% and the NBC classifier model produces the greatest accuracy value with the unigram with stop words model which produces an accuracy value of 0.78 or 78%. The following is a table of the results of the classification report comparison test data test from the two classifier models.

TABLE II. COMPARISON OF CNN AND NBC CLASSIFICATION REPORTS AGAINST DATA TEST

	Precision		**Recall**		**f1-Score**		**Support**	
	CNN	NB	CNN	NB	CNN	NB	CNN	NB
Netral	0.90	0.84	0.89	0.69	0.89	0.75	629	650
Poitive	0.89	0.77	0.87	0.83	0.88	0.80	659	627
Negative	0.85	0.75	0.89	0.82	0.87	0.78	669	680
Total	**0.88**	**0.78**	**0.88**	**0.78**	**0.88**	**0.78**	**1957**	**1957**

Based on the comparison results in Table II it can be seen that the results of precision and recall obtained from the CNN classifier model is 0.88 (88 %) while the precision results generated by the Naïve Bayes unigram with stop words classifier with 3000 features are 0.78 (78 %). These results indicate that the classifier model with Convolutional Neural Network algorithm can provide better accuracy results compared to the Naïve Bayes classifier model in sentiment analysis.

V. CONCLUSION AND FUTURE WORKS

The sentiment analysis conducted in this research uses English-language tweets obtained from Twitter using the Twitter API related to the topic "Turkey Crisis 2018". The whole sentiment analysis process starts from the data retrieval process, data classification with TextBlob which divides tweets into positive sentiments, negative sentiments, and neutral, and the use of Convolutional Neural Network and Naïve Bayes Classifier algorithms is done using the Python programming language.

The use of Deep Learning algorithm, Convolutional Neural Network in sentiment analysis has been successfully carried out in this research. The model architecture used in constructing this classifier model uses Keras Functional API model with the number of convolutional layers used is 3 layers with the addition of the kernel filter on each layer with the number 100 filters and kernel size will adjust the n-gram concept that will used in each convolutional layer, namely, bigram (2), trigram (3), and fourgram (4). The activation function used in the convolutional layer is ReLu. Three 1D max pooling layers are used in this model architecture to extract the maximum value from each filter. One fully connected layer with dropout is used to process the output from the max pooling layer with a total of 128 neurons. The output layer will consist of 3 neurons with the softmax activation function.

The CNN classifier model that has passed the training and evaluation process produces an accuracy value of 0.89 or 89% and in the test data testing process produces an accuracy of 0.88 or 88%. The accuracy results are then compared with the accuracy of the Naïve Bayes classifier model. This comparison of accuracy shows that the CNN classifier model has better accuracy values than the Naïve Bayes classifier model which produces an accuracy of 0.78 or 78%. From these results it can be concluded that the classifier model with Deep Learning algorithm produces better accuracy in sentiment analysis compared to the NBC classifier model.

Based on the results of the conclusions that have been described, it can be suggested that several things for further improvement and development include:

1) Retrieving tweet data from Twitter in greater numbers so that the classifier model can provide better accuracy in sentiment classification.

2) Comparing with other deep learning or machine learning algorithms.

3) The classifier model that has been built in this research is expected to be made into an application (front-end) either

desktop or website based to be utilized in analyzing sentiments on tweet data.

ACKNOWLEDGMENT

The research was conducted by the authors with the support of the Faculty of Computer Science and Information Technology in Gunadarma University, Depok, Indonesia.

REFERENCES

[1] F.Ratnawati and E. Winarko, "Sentiment Analysis of Movie Opinion in Twitter Using Dynamic Convolutional Neural Network Algorithm, Indonesian Journal of Computing and Cybernetics System (IJCCS), 2018, 12 (1), https://doi.org/10.22146/ijccs.19237

[2] Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint arXiv:1408.5882, 2014

[3] B. Liu, "Sentiment Analysis and Subjectivity," in Handbook of natural language processing, 2010, 2, 627-666

[4] A.A Sattikar and R.V. Kulkarni, "Natural Language Processing For Content Analysis in Social Networking," International Journal of Engineering Inventions, 2012, 1(4), 6-9

[5] Madcoms, Facebook, Twitter, dan Plurk dalam Satu Genggaman. Yogyakarta: ANDI, 2010.

[6] Y. LeCun, Y. Bengio, & G. Hinton, "Deep Learning," Nature International Journal of Science: doi:10.1038/nature14539, 2015, May 27.

[7] V. Bui and L.C. Chang , "Deep learning architectures for hard character classification," In Proceedings on the International Conference on Artificial Intelligence (ICAI) , 2016, p. 108.

[8] P. Devikar, "Transfer Learning for Image Classification of various dog breeds." International Journal of Advanced Research in Computer Engineering and Technology (IJARCET), 2016, 5(12), 2707-2715

[9] R. Refianti, A.B. Mutiara, and R.P. Priyandini, "Classification of Melanoma Skin Cancer Using Convolutional Neural Network," IJACSA, 2019, 10 (3), 409-417

[10] A. Esteva, B. Kuprel, R.A. Novoa, J. Ko, S.M. Swetter, H.M. Blau and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks", Nature, Vol. 542, pp 115-118, 2017

[11] T.J. Brinker, A. Hekler, J.S. Utikal, N.Grabe, D. Schadendorf, J. Klode, C. Berking, T. Steeb, A.H. Enk, and C.von Kalle, "Skin Cancer Classification Using Convolutional Neural Networks: Systematic Review", Journal of Medical Internet Research, Vol. 20, No.10, pp. 11936- 11946, 2018