Twitter User Sentiment Analysis for RUU Omnibuslaw Using Convolutional Neural Network

Popon Dauni, Muhammad Ali Ramdhani,
Dimas Ramdhani Suryapratama, Wildan Budiawan Zulfikar and
Jumadi

April 18, 2021

Twitter User Sentiment Analysis For RUU Omnibuslaw Using Convolutional Neural Network

Popon Dauni
Departement of Informatic Engineering
Faculty of Science and Technology
UIN Sunan Gunung Djati, Universitas Kebangsaan
Bandung, Indonesia
popon.dauni@if.uinsgd.ac.id

Muhammad Ali Ramdhani
Departement of Informatic Engineering
Faculty of Science and Technology
UIN Sunan Gunung Djati
Bandung, Indonesia
m_ali_ramdhani@uinsgd.ac.id

Dimas Ramdhani Suryapratama
Departement of Informatic Engineering
Faculty of Science and Technology
UIN Sunan Gunung Djati
Bandung, Indonesia
1167050052@student.uinsgd.ac.id

Wildan Budiawan Zulfikar
Departement of Informatic Engineering
Faculty of Science and Technology
UIN Sunan Gunung Djati
Bandung, Indonesia
wildan.b@uinsgd.ac.id

Jumadi
Departement of Informatic Engineering
Faculty of Science and Technology
UIN Sunan Gunung Djati
Bandung, Indonesia
jumadi@if.uinsgd.ac.id

Abstract*—The general function of social media is for online interaction with many people. Moreover, social media have functions for sharing information, discussion, and giving an opinion media about some topics that a lot of people talk about, one of that media is Twitter. An atopic will show many opinions and different responses from everyone. This study was for making an analysis opinion from social media Twitter user about Rancangan Undang-Undang Omnibuslaw topic using a Convolutional Neural Network method wich one of Deep Learning method. This study has been done a sentiment analysis with opinion data from many different people through the tweet they making, Preprocessing and weighting are done using Word2vec which give 84% result accuracy of an algorithm from 10-time testing. Based on 2.820 tweet data, the result is 1.320 data of positive sentiment, and 1.500 data of negative response for the Rancangan Undang-Undang Omnibuslaw topic in Indonesia.

***Keywords*—Twitter, Sentiment Analysis, Convolutional Neural Network (CNN).**

I. Introduction

In the beginning of 2020, Dewan Perwakilan Rakyat Republik Indonesia (DPRRI) received submissions for the Omnibuslaw Draft (RUU) from the government of the Republic of Indonesia which has two categories, namely the RUU Omnibuslaw Cipta Lapangan Kerja and RUU Omnibuslaw Perpajakan. Omnibuslaw is a term for rules that include various categories. Omnibuslaw is defined as the law for all which comes from the Latin omnis which means "many" or "for all" [1].

With the submission of the draft Omnibuslaw, various responses emerged from various circles of society. One of the omnibuslaw drafts that have generated many opinions and criticisms from the public is the draft Omnibuslaw Bill on Job Creation, especially from those who feel directly affected if the Omnibuslaw draft is enacted.

Along with the development of information media, criticism and support from the public regarding the Omnibuslaw Bill were mostly conveyed through social media. One of the media used to convey opinions regarding the Omnibuslaw is Twitter social media, especially when the topic of the Omnibuslaw Bill is being discussed, many are sharing opinions and news on a real-time basis via Twitter social media so that there are more and more opinion data.

The Twitter social media platform makes it easy to access existing Tweet data by registering for API access through the Twitter Developer service. After getting API access, tweet data can also be accessed with the help of a web scraping service so that tweet data can be easily obtained for various purposes, one of which is for research purposes.

With the explanation above, an analysis is needed that can show the direction of the existing opinion. With the hope that the public can find out the direction of opinions conveyed on Twitter social media through a scientific approach so that the flow of existing opinions has information that is easier to understand and does not make people confused in receiving existing information. The algorithm used in this study is the Convolutional Neural Network algorithm which will play a role in the classification stage of the opinion data that has been obtained. By also applying the Deep Learning method so that the classification process carried out by the system has a good accuracy value because the existing data will go through various classification detection processes.

II. Theoretical Basis

A. Convolutional Neural Network

The convolutional neural network is defined as an algorithm that is usually used for processing image and text data that is included in the category of a neural network algorithm. The word convolution itself is defined as a matrix that functions to classify and filter images and text [2].

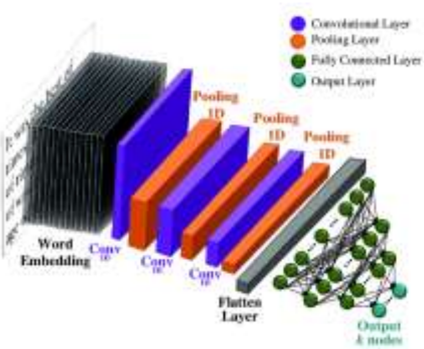

Figure 1. Convolutional Neural Network Architecture [3]

There are several layers used in the Convolutional Neural Network algorithm that function as filters in each process called the training process [2].

In the training process, there are 3 stages, namely the Convolutional layer, the pooling layer and, the fully connected layer.

From each layer contained in the convolutional neural network architecture, the neurons will be linked to the next layer. The last layer will display the output in the form of classified data from the previously connected layers [3].

B. DeepLearning

Deep learning is an application of artificial neural networks that mimic the workings of the human cortex which has many hidden layers and is included in the study of Machine Learning which functions to provide accuracy in various studies such as detecting an object. Deep learning can automatically process data such as images or text without having to recognize input from humans, in contrast to traditional Machine Learning which must first recognize input [4].

Deep learning algorithms have features that can extract automatically the problem-solving process. Algorithms like this are indispensable in artificial intelligence because they can reduce the burden on the program in processing a problem [5]. Solving problems carried out by Deep Learning on a computer system applies the concept of hierarchy. This hierarchical concept combines simple concepts in learning a more complex concept. Deep learning can learn from complex mapping functions, from input to output, it does not require concepts or artificial input from humans [6].

C. Text Preprocessing

Text preprocessing is one of the steps in text mining in which it receives information with imperfect text data structures. In-text preprocessing, there will be several processes that make information that previously had an imperfect text data structure extracted into information that has a more perfect data structure than before.

After going through the stopword process, the words that have been selected will be processed to remove the affixes in the steaming process, the deleted words will return to their basic word form without changing the word group [7].

D. Confusion Matrix

A confusion matrix is a calculation method for the classification process in the concept of data mining to find out how data can be classified correctly [8]. In the configuration matrix classification, four terms are known for the classification results of the configuration matrix, including:

1) TP (True Positive) is data that is detected correctly and is positive.
2) TN (True Negative) is data that is detected correctly but is negative.
3) FP (False Positive) is data that is detected incorrectly but is positive.
4) FN (False Negative) is data that is detected as false and is negative.

In general, the classification calculations included in the configuration matrix consist of recall, precision, F1-score, accuracy, and support [9].

A recall is a calculation of the description of the success rate of a system when recovering information. Recall has an equation formula as in equation (1) below:

$$recall = \frac{TP}{TP + FN} \qquad (1)$$

Precision is the level of accuracy between the answers displayed by the system and the information expected by system users. Precision has an equation formula as in equation (2) below:

$$precision = \frac{TP}{TP + FP} \qquad (2)$$

The F1-score or F-measure is a calculation used as a classification evaluation of the combined precision and recall. When a case of precision and recall have different values, the F1-score becomes the reciprocal value between precision and recall with the weighted harmonic values of the mean precision and recall [10]. The F1-score has the equation formula as shown in equation (3) below:

$$F_1 = 2\,x\,\frac{precision\ x\ recall}{precision + recall} \qquad (3)$$

Accuracy is the value of accuracy in a classification process. The accuracy value comes from the distribution of the results of the entire classification that is correct and the amount of data that has been classified. Accuracy has the same formula as in equation (4) below:

$$Accuracy = 100\%\,x\,\frac{True\ Classification\ Total}{Classification\ Total} \qquad (4)$$

Support is the total value of all documents classified with a prediction of true or false classifications. Support can be used to determine the number of documents that appear [11].

III. Research Methodology

The research for this final project used several methods which are data analysis method, and research method.

A. Data Analysis Method

The data obtained from Twitter is raw data or unstructured data, so the data must go through the data processing stage so that later the data to be processed into information has the same data structure so that it can produce information as expected.

The data processing technique used in this study is a descriptive analysis technique by analyzing the quantitative data that has been obtained to conclude the analyzed data [12].

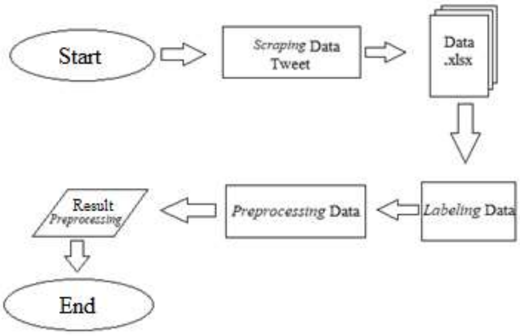

Figure 2. Data Processing Process Flow

Before the data is ready to be used for this research, the data that has been obtained from the scraping process will go through the data labeling stage, then the data will be processed first with the preprocessing process so that the existing data is data free from noise data. After the data is declared free from noise. Preprocessing stages using libraries available in the Python programming language are carried out in several stages, namely: lowercase, tokenizing, stemming, stopword, and regular expression.

B. Research Method

After the data is declared free from noise, then the preprocessing data will go through the next research stage, namely the data embedding stage to convert existing data into vector form so that the data can be processed by the system [13].

The embedding token stage is carried out with the help of a library that is entered into the Python language program which is used with a data dictionary source obtained from the Corpus Wikipedia to change the shape of the vector into a number whose size is even smaller to accommodate more information [8].

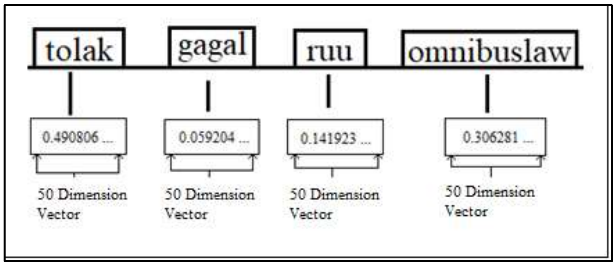

Figure 3. Example of Converting Words into Vector

After the tweet data becomes a vector form that can be read by the system, the vector dataset is divided into a test dataset and a practice dataset according to the given division rules.

After the test and training dataset is determined, the system reads the input test data vectors and makes predictions from the training dataset that has been studied.

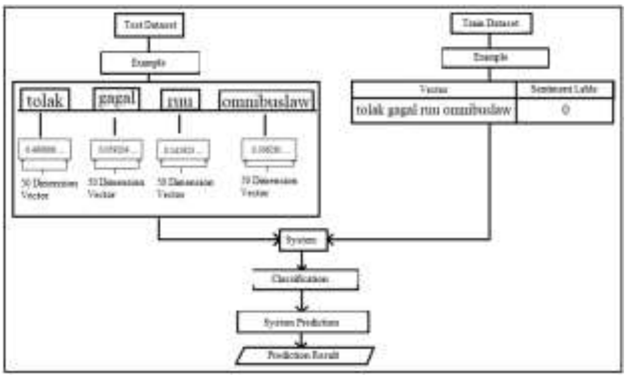

Figure 4. Example of Dataset Stage into the System

The stages of explaining the process described in Figure 4 are as follows:

1. The system accepts input in the form of a token vector dataset with categories of training data and test data.
2. When the classification process begins, each vector data originating from the test data is read by the system, then each vector token is matched with the training data that has been studied to classify the data with the help of the existing vector data dictionary.
3. After finding a match, the system repeats for each vector that is read as much as the given test data
4. After all the classification processes have been completed, the system will provide the predictive results of the classification made by the system based on the training data parameters provided and match the classification results with the actual dataset and represent the results.

The system algorithm process is described as shown below

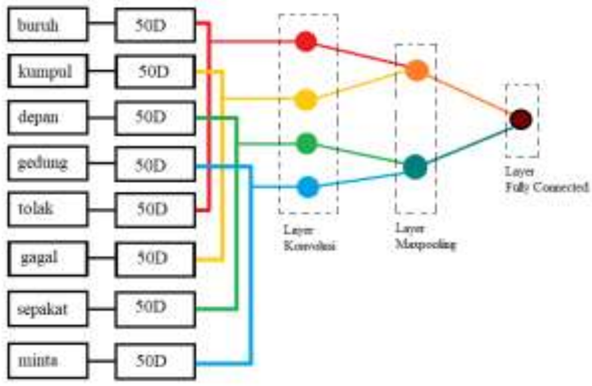

Figure 5. Illustration of CNN Model

In the classification process of the Convolutional Neural Network algorithm, four main layers function as data processors when the data is clean from noise as a result of the preprocessing stage. The four main layers are:

1. Embedding Layer

   This embedding layer contains words that are stored in a low-dimensional vector. In this layer, the words are converted into an array of tokens which are represented in a sentencing matrix. At this layer, all data will have the same format, which is in the form of vectors that aims to simplify the system in carrying out the classification process.

TABLE I EMBBEDING LAYER

Word	buruh	kumpul	depan	gedung	tolak	gagal	sepakat
Value	0.686	0.444	-0.46	-0.42	0.978	0.892	0.492

2. Convolutional Layer
The next layer is the convolution layer which functions as a vector filter to determine the vector with the highest value. The highest value is taken from each word vector to facilitate the pooling stage to find word weights that have more meaning so that when entering the fully connected layer, the vector will produce relevant information.

TABLE II CONVOLUTIONAL LAYER

Word	buruh	kumpul	depan	gedung	tolak	gagal	sepakat
Convolution	0.68	0.44	-0.46	-0.42	0.97	0.89	0.49

3. MaxPooling Layer
In this pooling layer, the convoluted vector is then taken the highest value from each feature map so that it can then be merged into the next layer and has a vector that contains good information.

TABLE III MAXPOOLING LAYER

Word	buruh	tolak	gagal	minta
Maxpooling	0.68	0.97	0.89	0.79

4. Fully Connected Layer
The fully connected layer contains connected vectors because it is considered to have the highest weight vector of the previous layer process and is the layer that produces system output [14].

TABLE IV FULLY CONNECTED LAYER

Word	tolak
Fully Connected	0.97

From the illustrative example above, it can be seen that the training data is in the form of a tweet sentence "workers gathered in front of the DPRD building call to reject the omnibuslaw and for the DPRD to thwart the omnibuslaw base on the labor request agreement" labeled sentiment 0 has the word reject as the word fully connected. This makes the word "reject" learned by the system as a word with 0 sentiments. So that when given the test data that has the word "reject" it will be predicted as a sentiment labeled 0.

IV. DESIGN, SIMULATION, AND REALIZATION

The research implementation of the system design process that has been carried out includes several aspects, namely the implementation of data collection, the implementation of data labeling, the implementation of preprocessing, and the implementation of word weighting.

A. Preprocessing

Implementation of Data Preprocessing in this study was carried out in several stages, namely Lowercase, Tokenizing, Stemming, Stopword, and Regular Expression.

```
tkr = RegexpTokenizer('[a-zA-Z@]+')

factory = StemmerFactory()

stemmer = factory.create_stemmer()

factory = StopWordRemoverFactory()

stopword = factory.create_stop_word_remover()

tweets_split = []

for i, line in enumerate(tweets):

    tweet = str(line).lower().split()     1

    tweet = stemmer.stem(str(tweet))      2

    tweet = stopword.remove(str(tweet))   3

    tweet = tkr.tokenize(str(tweet))      4

  tweets_split.append(tweet)

print(tweets_split[3])
```

```
['omnibuslaw', 'untung', 'nelayan', 'kecil']
```

Figure 6. Data Preprocessing Command

As seen in the picture above, command number 1 is the command for the Lowercase and Tokenizing process. Then command number 2 is the command for the Stemming process, command number 3 is for the Stopword process. And command number 4 is the command for the RegularExpression.

B. Embedding

The implementation of the word weighting stage or data weighting in this study was carried out using the Word2vec library. Using Word2vec in the system as shown in the picture below.

```
w2vModel =
word2vec.KeyedVectors.load_word2vec_format("glove_wiki_id_50_to_word2
vec.txt", binary=False, limit=30000)

tokenizer = Tokenizer()

tokenizer.fit_on_texts(tweets_split)

X = tokenizer.texts_to_sequences(tweets_split)

maxlentweet = 100

X = pad_sequences(X, maxlen=maxlentweet)

print(X.shape)

w2vModel.wv['setuju']
```

```
array([-1.264637,  0.963529,  0.769047, -0.992108,  0.240636, -1.61337 ,
        0.945135, -0.073607, -0.125121, -0.039511, -0.034403, -0.425573,
       -0.511393, -0.115798, -0.428   , -0.51261 , -0.316265,  0.546554,
       -0.40907 ,  0.950288,  0.338239,  0.101869, -0.36186 ,  0.257527,
       -0.070514, -1.087531, -0.64575 ,  0.119138, -0.228568,  0.050485,
       -0.174049, -0.298303,  0.070064,  1.735978,  0.421546, -0.498136,
        0.291551, -0.348009,  0.345556, -0.256997, -0.32094 ,  0.236952,
        0.458261, -0.396618, -0.230204, -0.1698  , -0.229055,  0.474114,
       -0.595608,  0.8218  ], dtype=float32)
```

Figure 7. Use of the Word2vec Command

The picture above is an illustration of using the word weighting process for the word "agree". It can be seen that the word "agree" has a vector weight of 50 dimensions with the use of a 1-dimensional array.
Another word weighting illustration is shown in the table below.

TABLE V EMBEDDING WORD

Word	Value
"buruh"	[-0.562035, 0.431941, 1.076237, 0.43465 , -1.002631, -1 .118669, 0.140762, 0.781447, -0.854622, -0.646436, -1.2340 82, 0.175297, 0.226949, -0.081893, -0.7865 , 0.396706, -0.7193 65, 1.020527, -0.493448, 1.194103, -0.581254, -0.692198, 0.7852 09, 0.573736, -0.108268, -1.760587, -0.749763, -0.469644, 0.1630 16, 0.904432, 1.248905, -0.884914, -0.165467, 0.432833, -0.1965 96, -0.046668, 0.571956, 0.204011, 0.932222, -0.159791, -0.7033 98, 0.053933, -0.002418, 0.691289, 0.560941, 0.40289 , 0.3159 02, -0.378922, -0.816211, 0.882193]
"tolak"	[0.490806, 0.407934, 0.563779, 0.483614, 0.479262, -0 .535606, -0.167378, 0.146994, 0.62842 , 0.889968, 0.1161 93, 0.011636, -0.159768, -0.303885, 0.20616 , -0.197957, -0.8990 88, 0.101876, -1.073625, 0.607658, -1.700634, -0.113661, -0.7671 97, 1.046719, -0.022405, -0.8693 , 0.486589, 0.541174, -0.2694 1 , 1.116227, -0.447768, -0.438652, -0.303308, 0.058025, -0.1126 48, 0.111414, 0.794698, -0.234784, 0.028134, -0.632644, 0.3193 38, -0.442331, -0.701739, 0.067139, -0.121346, 0.02138 , -0.2784 46, 1.081892, -0.268877, -0.404289]
"sepakat"	[-0.550309, 0.930404, 0.545297, -0.797978, -0.026297, -0 .923739, 0.913884, -0.038695, -0.686749, 0.347401, -0.1239 4 , -0.575179, 0.285596, -0.023983, -0.351622, -0.886999, -0.1934 54, 0.742572, -0.296869, 0.983796, 0.201401, -0.097716, -0.1854 71, 0.59912 , -0.185104, -0.938373, -0.602434, 0.155587, -0.1927 61, 0.015987, 0.250276, -0.528425, -0.345295, 1.100533, -0.1523 18, -0.579499, 0.223967, -0.339166, 0.309802, -0.035333, -0.8961 84, 0.409224, 0.354923, -0.297973, -0.224653, -0.053027, -0.1860 45, 1.02099 , -0.465026, 0.349276]

C. Simulation

At the simulation stage, the system was tested 10 times and by setting the epoch value or algorithm cycle in studying the entire dataset given as 10 and 20. As well as dividing the test data and training data by 5 variations, namely, 90% of the training data with 10% of the test data. , 80% training data with 20% test data, 70% training data with 30% test data, 60% training data with 40% test data, 50% training data with 50% test data.
In the first simulation, using an epoch of 10 and by using the distribution of the dataset by 90% the training data were 2,538 data with 10% test data as many as 282 data resulting in a configuration matrix value as seen in Figure

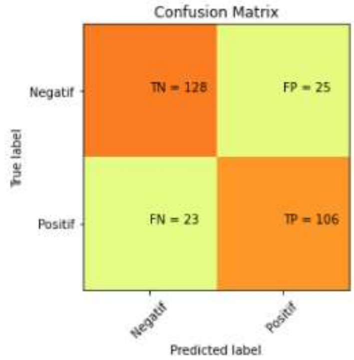

Figure 8. First simulation Confusion Matrix

The confusion matrix test value I above have the calculation of precision, recall, f1 score, and accuracy using equations (1), (2), (3), (4) for positive sentiment as follows:

$$\text{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{FP+TP}}\ \boldsymbol{x}\ \mathbf{100}\% = \mathbf{81}\%$$

$$\text{Recall} = \frac{\boldsymbol{FP}}{\boldsymbol{FN+TP}}\ \boldsymbol{x}\ \mathbf{100}\% = \mathbf{82}\%$$

$$\text{F1-Score} = 2 \text{ x } \frac{\boldsymbol{recall\ x\ precision}}{\boldsymbol{recall+precision}} = \mathbf{82}\%$$

The calculation for negative sentiment is as follows:

$$\text{Precision} = \frac{\boldsymbol{TN}}{\boldsymbol{FN+TN}}\ \boldsymbol{x}\ \mathbf{100}\% = \mathbf{85}\%$$

$$\text{Recall} = \frac{\boldsymbol{TN}}{\boldsymbol{FP+TN}}\ \boldsymbol{x}\ \mathbf{100}\% = \mathbf{84}\%$$

$$\text{F1-Score} = 2 \text{ x } \frac{\boldsymbol{recall\ x\ precision}}{\boldsymbol{recall+precision}} = \mathbf{84}\%$$

$$\text{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}}\ \boldsymbol{x}\ \mathbf{100}\% =$$

$$\frac{\mathbf{106+128}}{\mathbf{106+128+25+23}}\ \boldsymbol{x}\ \mathbf{100}\%$$

$$= \mathbf{83}\ \%$$

D. Result of Realization

After ten-time tests, the results are as shown in the table below:

TABLE VI EPOCH VALUE 10

Training data		Testing data		Accuracy (%)
Percentage	Data	Percentage	Data	
90%	2.538	10%	282	83
80%	2.256	20%	564	79
70%	1.974	30%	846	78
60%	1.692	40%	1.128	77
50%	1.410	50%	1.410	76
Average Percentage				78,6

TABLE VII EPOCH VALUE 20

Training data		Testing data		Accuracy (%)
Percentage	Data	Percentage	Data	
90%	2.538	10%	282	90
80%	2.256	20%	564	88
70%	1.974	30%	846	87
60%	1.692	40%	1.128	87
50%	1.410	50%	1.410	82
Average Percentage				86,8

From the evaluation data displayed based on the table above, it can be said that using training data of 2,538 data has the highest accuracy value compared to other tests in the use of epoch of 10. And it can be seen from five tests, it has an average value of 78, 6%.

Whereas with the use of an epoch value of 20, the highest accuracy value is in the percentage of using training data of 90% with data as much as 2,538 and, from five tests, an average accuracy of 86.8% is obtained.

E. Representation

From the two tables' result of realization, it can be seen that the level of accuracy for all divisions based on the amount of data has a decrease when the amount of training data is less than the previous percentage as illustrated in the accuracy percentage graph below.

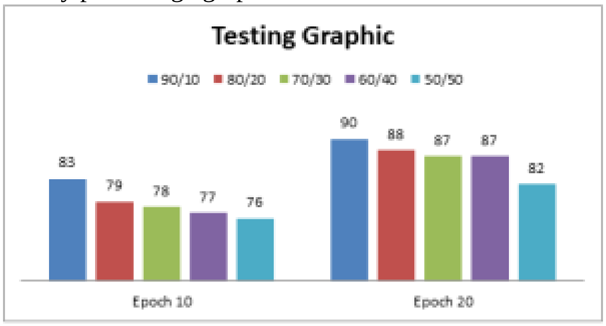

Figure 9. Percentage Graph of Testing Accuracy

The dataset used is a dataset taken within 6 months. And each of them has a different amount of data each month.

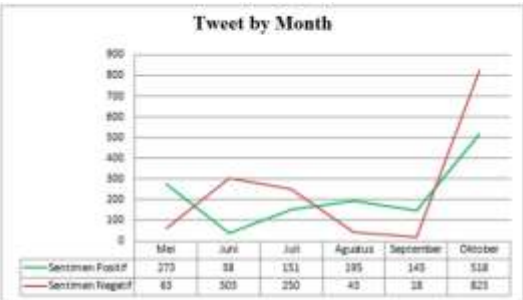

Figure 10. Tweet Sentiment Graph by Month

From the graph above, it can be said that in a period of 5 months, namely in May, June, July, August, and September, each positive and negative sentiment has varied results with an up and down graph. Meanwhile, in the last month, October, the number of sentiment data has increase, namely 823 data for negative sentiment and 518 data for positive sentiment.

The increase in the amount of data that occurred in October was partly due to the passing of the Omnibuslaw Bill by the DPR RI on October 5, 2020. This triggered the public to express their opinion through Twitter social media, which was even more so than the previous month.

Based on the data retrieval process using the keywords "Omnibuslaw", "Ciptaker", and "Job Creation", each keyword has a different number of search results for the entire dataset used. The number and comparison can be seen in the graphic image below.

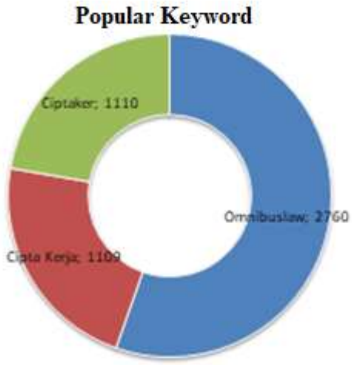

Figure 11. Popular keyword chart

Based on the graph above, it can be seen that the keyword "Omnibuslaw" has the highest number with 2,760 mentions in the dataset used. Meanwhile, the keyword "Ciptaker" is mentioned 1,110 times and the keyword "Job Creation" is mentioned 1,109 times.

The sentence or opinion data analyzed and processed by the system has several words that appear frequently. Usually, it is a representation of sentiments that appear more frequently or exist in the dataset used [12]. The words that often appear can be made into a visual form of a wordcloud image. The font size in the wordcloud represents the number or frequency of the word appearing in the sentiment dataset. As seen in the image below

Figure 12. Wordcloud Positive (A), And Negative (B)

As can be seen in the wordcloud picture above, in the positive sentiment class that is mostly found are the words “Omnibuslaw”, “Cipta”, “Kerja”, “Omnibus”, “Law”, “For”, “RUU”, “workers” indicating that the word is widely used in tweets that are positive sentiments. As for the words "Reject", "RUU", "Omnibus", "Law", "DPR", "Cipta", "Kerja", and many other words are found in negative sentiment tweets.

F. Common Error Analysis

The level of accuracy does not reach 100% due to several factors such as

- Using local language
- Weak dataset
- Weak labeling
- Fail at preprocessing

V. CONCLUSIONS AND SUGGESTIONS

A. Conclusions

From the tests carried out using the Convolutional Neural Network algorithm in text classification in this study, the algorithm can work well in classifying text. From the tests carried out, the highest level of algorithm accuracy with a value of 86.8% is in the distribution of training data and test data with a percentage of 90%: 10%. And the comparison of using epoch 10 with epoch 20 results in a higher classification accuracy at the use of an epoch value of 20. With the use of a dataset of 2,820 data, every reduction in the amount of training data, the classification results also decrease. It can be concluded that the more training data used and the more often the data is trained (epoch), the higher the accuracy rate will be and is by concept of deep learning.

The level of accuracy does not reach 100% due to several factors such as the use of regional languages, the use of abbreviated sentences so that they are inappropriate during the preprocessing process, failure in the filtering process in preprocessing can cause the trained data to be confused or inappropriate. Apart from mismatches during the preprocessing process, the cause of low accuracy or not reaching 100% is due to the dataset itself. Including the data that should have a positive meaning, but the algorithm reads it becomes a negative meaning and vice versa [15]. This can also affect the accuracy of the algorithm in carrying out the classification process and predicting sentiment during the dataset testing stage.

In addition to producing a high score accuracy of sentiment classification, the research that has been conducted has also resulted in Twitter user reviews regarding the topic of the Omnibuslaw Bill with a greater number of negative sentiments with the words "Reject", "RUU", "Omnibus", "Law" sentiment. and "CiptaKerja" compared to the number of positive sentiment reviews with the sentiment of the words "Omnibuslaw", "For", "Worker" in the last 6 months using the keywords "Omnibuslaw", "Ciptaker" and "Cipta Kerja" that have been obtained.

B. Suggestions

Based on the research that has been done, several suggestions can be considered in the next stage of development:

1. Increase the number of datasets used so that the results obtained are more real and better labeling can increase the value of research accuracy so that at the time of the dataset testing process, there is no data that the system cannot recognize.
2. Collaborate research models to increase the accuracy can be combined with LSTM architectural models, one of which is the Recurrent Neural Network (RNN) model.
3. To get a more original opinion dataset, especially when researching on political topics it is hoped that there will be an analysis of fake accounts or an analysis of opinions conveyed by programmed robot accounts. This is necessary so that there is no concept of opinion manipulation by robot accounts for certain purposes

REFERENCES

[1] F. F. Busroh, "Conceptualization of Omnibus Law in Solving Land Regulation Problems," *Arena Hukum* , vol. 10, no. 2, pp. 227-250, Agustus 2017.

[2] G. A. A. Santoso, "Implementation of Hard-Based Deep Learning for Face Recognition," *Jurnal Emitor* , vol. 18, no. 01, pp. 15-21, 2018.

[3] K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. B. K. Kowsari, "Text Classification Algorithms: A Survey," *Information*, pp. 2-68, April 2019.

[4] Y. Bengio, G. H. Y. LeCun, "Deep learning," *Nature*, vol. 521, pp. 436-444, Mei 2015.

[5] B. Irawan, C. S. B. P. Putra, "Hatespeech Detection Using Convolutional Neural Network Algorithm Based On Image," *e-Proceeding of Engineering* , vol. 5, no. 2, p. 2395, Agustus 2018.

[6] Taweh Beysolow II, Introduction to Deep Learning Using R : A Step-by-Step Guide to Learning and Implementing Deep Learning Models Using R, 1st ed.: Apress, 2017.

[7] P. Pande Yudiastra Gusti Ngurah Mega Nata, "Preprocessing Text Mining Pada Email Box Berbahasa Indonesia ," *Konferensi Nasional Sistem & Informatika 2017* , pp. 479-483, Agustus 2017.

[8] A. Azman, M. A. Eissa M.Alshari, "Improvement of Sentiment Analysis based on Clustering of Word2Vec Features," 28th International Workshop on Database and Expert Systems Applications, vol. DOI 10.1109, no. DEXA.2017.41, pp. 123-126, Agustus 2017.

[9] Arif Bijaksana I.L.S. Nabila Nanda Widyastuti, "Word2vec Analysis for Calculations," e-Proceeding of Engineering , vol. 5, no. 3, pp. 7603-7612, Desember 2018.

[10] C. Josephine Christ A. K. Santra, "Genetic Algorithm and Confusion Matrix for Document Clustering ," International Journal of Computer Science, vol. 9, no. 1, pp. 322-328, Januari 2012.

[11] Triowali Rosandy, "*Perbandingan Metode Naive Bayes Classifier Dengan Metode Decision Tree (C4.5) Untuk Menganalisa Kelancaran Pembiayaan* (Study Kasus : KSPPS / BMT AL-FADHILA) ," *TIM Darmajaya* , vol. 02, no. 01, pp. 52-62, Mei 2016.

[12] Kemas Muslim L. Brata Mas Pintoko, "Sentiment Analysis of Online Transportation Services on Twitter Using the Naïve Bayes Classifier Method," e-Proceeding of Engineering , vol. 5, no. 3, p. 8121, Desember 2018..

[13] G. A. Buntoro, " Sentiment Analysis of Candidates for Governor of DKI Jakarta 2017 on Twitter," *Integer Journal*, vol. 2, no. 1, pp. 32-41, Maret 2017.

[14] Y. S. Fransiska, " Twitter Sentiment Analysis for Movie Reviews Using the Naive Bayes Classifier (NBC) Algorithm on Sentiment R Programming," Jurnal Siliwangi , vol. 5, no. 2, 2019.

[15] A. Putra, " Application of Omnibus Law in Regulatory Reform Efforts," Jurnal Legislasi Indonesia, vol. 17, no. 1, pp. 1-10, Maret 2020.