JURNAL EMACS
[Engineering, MAthematics and Computer Science] Vol.2 No.1 January 2020: 29-34 e-ISSN: 2686-2573

Multilabel Classifcation for Toxic Comments in Indonesian

Reinert Yosua Rumagit

Computer Science Department, School of Computer Science,
Bina Nusantara University,
Jakarta, Indonesia 11480
reinert.rumagit@binus.edu

Abstract – *The more rapid development of the internet world, users can make comments on a variety of content on social networks, such as social media, blogs and others. Free users make comments triggering negative comments, making insults and incitement. By classifying user comments it is hoped that the system can be smarter to be able to distinguish threat, insult and incitement comments. The technique for classifying user comments uses deep learning, consisting of 6 classes. The results of experiments that have been conducted show that deep learning models produce an accuracy rate above 98%.*

Keywords: *Classification; toxic comment; deep learning.*

I. INTRODUCTION

The development of the internet is very fast making many social networking sites popping up and growing, various types of messengers continue to emerge. This allows users to freely send messages and comment on social networks. According to Dreamgrow (Kallas, 2019), the largest social networking user is occupied by Facebook with a total of more than 2 billion users, then followed by Youtube and Instagram. The increasing number of users from social media means that the amount of content will increase. Moreover, social media users who make their content interesting tend to want to be responded to or received recognition from other users, whether it's in the form of likes or making comments. That way the number of comments will be more and more. Sometimes the comments and open discussion can trigger debate, can be due to differences of opinion or because they are upset with the content presented. But often the debates that occur appear bad things and use dirty methods to debate. Dirty ways can cause a big fight on social media, so using toxic comments to do the offensive.

Toxic comments can contain words of threat, obscene, insult or hatred of identity, so that it will create harassment on social media, or commonly called online harassment. As a result of these acts of harassment, some people will stop giving opinions or try to avoid debates on social media that result in unhealthy and unfair discussions. So that social networking platforms and online communities find it very difficult to facilitate fair conversations and people do not feel restricted from making comments or to turning off the user comments feature. All of which aims to keep online conversations constructive and inclusive, that is what the provider wants. The automatic classification of toxic comments, such as expressions of hatred, threats and insults, can help keep discussions just right, fair and useful. This study focuses on building models using machine learning to be able to detect conversations or comments that contain toxicity such as, threats, obscenity, insults and hatred of certain identities in the Indonesian language Text.

Related Work

Text processing is the most important thing for managing text in order to provide useful information. Utilization of text processing has been done a lot, such as text processing to summarize documents (Rumagit, Setiyawati, & Bangkalang, 2019). Text processing is also carried out to classify documents, as in research conducted by Reinert et al. Which classifies the personality of Facebook social media based on user posts (Rumagit & Girsang, 2018). Classification of toxic comments has also been done a lot, such as research conducted by Mestry et al. Where they conducted toxic classifications using the CNN algorithm and the Fast Text method for word embedding (Mestry, Bisht, Chauhan, Tiwari, & Singh, 2019). Research conducted by Georgakopoulos et al is to classify toxic comments using CNN (Georgakopoulos, Vrahatis, Tasoulis, & Plagianakos, 2018). Srivastava et al in their research identified toxic comments using the Capsule Network (Srivastava, Khurana, & Tewari, 2018). Research related

to the classification of toxic comments was also conducted by Mujahaed et al using logistic regression and neural networks models (Saif, Medvedev, & Medvedev, 2018). As for Patrick et al in his study also conducted a classification of toxic comments using machine learning methods namely logistic regression (Ozoh, M O, & Adigun, 2019). Sharma also conducted research on the classification of toxic comments using machine learning and neural networks, the Convolutional Neural Network (Sharma & Patel, 2018). The research on toxic classification carried out by Betty et al is doing toxic comment classification using Logistic Regression and LSTM (Aken, Risch, Krestel, & Alexander, 2018).

II. METHOD

A. The Concepts

1. The concepts of methodology in this study are shown in Figure 1. Based on the illustrations in Figure 1, the core steps of this study are as follows. Data Construction, this stage includes the stage of crawling data from social media and preprocessing.
2. Word & Document Representation, at this stage includes the stages of making words models and making LSTM models.
3. Toxic Classification & Evaluation, this stage includes sigmoid classifier and evaluation.

Detailed explanation of each step will be discussed in the next section.

B. Data Construction

At this stage, all comment data and status from social media users are collected. The process of retrieving data from social media uses the social media API. The social media used are Facebook and Twitter, and will be the testing data in this research. The training data used in this study uses a dataset provided by Kaggle (Google, 2018). The dataset contains 150k comments that have been labeled, the data set is then translated into Indonesian using the python library. Figure 2 shows the distribution of the dataset that will be used as training data in this study.

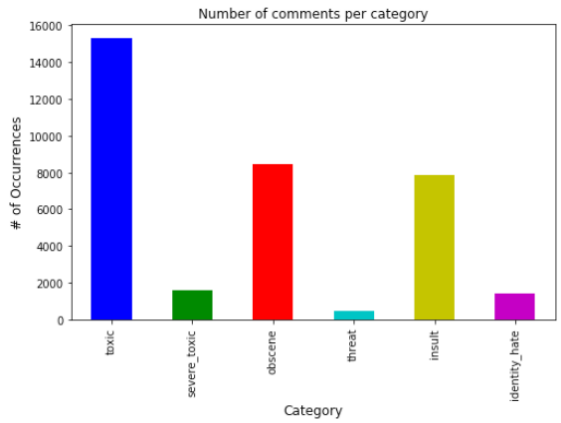

Figure 2 Distribution Dataset

Text preprocessing is where the process of cleaning up the data to be able to ensure the data to be used is consistent and uniform.

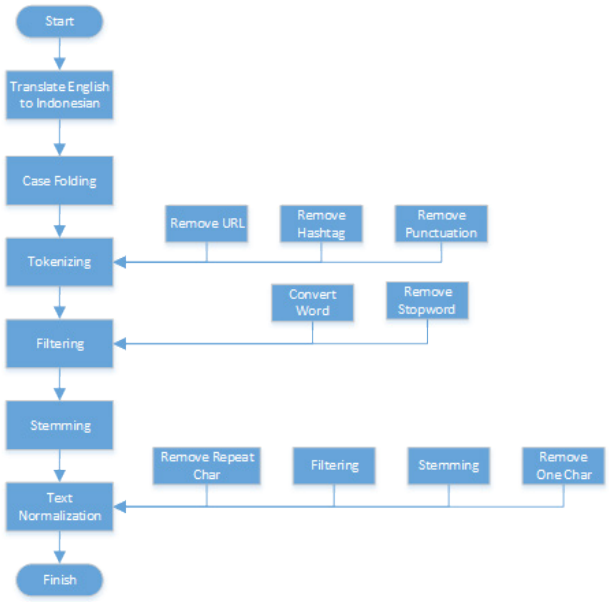

Figure 3 Step of Preprocessing

As for the several preprocessing stages carried out in this study, these stages are shown in Figure 3.

1. *Translate English to Indonesia*, at this stage all words will be converted into Indonesian using the Python library.
2. *Case Folding,* this stage will change all capital letters of the word found into lowercase letters. For example, it is shown in the sentence below..

   "Aku benci sama Kamu"

   would be changed into:

   "aku benci sama kamu".
3. *Tokenizing*, this stage will delete all URLs, hashtags and punctuation found in the sentence and do the separation of all sentences into words. An example is shown in the sentence below.

   "mungkin bisa di bagi ke teman, saudara dkk alamat survei: https://docs.google.com/forms/d/12gzlkbuzwsdssmxli6xv86kn8uzvqwrnvjsimd7ddcw/viewform"

   would be changed into:

   mungkin
   bisa
   di
   bagi
   ke
   teman
   saudara
   dkk
   alamat
   survei".
4. *Filtering*, is the stage where to remove words that are considered meaningless using the stopword list from Tala (Tala, n.d.). This stage also involves the process of converting non-standard words into *Kamus Besar Bahasa Indonesia* (Great Dictionary of the Indonesian Language). An example can be

seen in the sentence below.

"*akhirnya nemuin makanan ini gak berhenti makan deh*"

would be changed into:

"*menemukan makanan tidak berhenti makan deh*".

5. *Stemming*, at the stemming stage, it is done to eliminate affixes at the beginning of a word such as me-, ber-, ter-, and so on or an affix at the end of a word such as -kan, -an, -i, and so on. An example can be seen in the sentence below.

"*menemukan makanan berhenti makan deh , makan adalah sebuah talenta*"

would be converted into:

"*temu makan henti makan deh makan adalah sebuah talenta*".

6. *Normalization*, is the final stage of the preprocessing sequence, where at this stage the erasure of the repeated letters will be done which will cause the wording to be disorganized or structured. After that the process is continued by repeating the Filtering and Stemming steps, to ensure that the word is not in Stopword and has been removed. The very end of this step will delete the character 1 (one) letter that has no meaning. An example can be seen in the sentence below.

"*malam Agus gilaaaaaaa kerennnnn Baaaaaangeeeeet maju kocak a*"

would be converted into:

"*malam michael gila keren sekali maju lucu*".

C. Word & Document representation

At this stage will explain how to represent words into a machine-trained model.

1. Word Representation
   The basic idea of a word representation model is by mapping words into high-dimensional dimensional vectors. Where the distance between each word in the vector / space depends on the similarity of the context. In this study, each word that appears will be entered into a dictionary of words (Bag of Words), the words will be converted into a number and then the words that have been changed will be placed in a vector. In this study the number of attribute features that have been extracted is 251,097 words. The entire extraction of word features is stored in a file with pickle format.
2. Document Representation
   At this stage the modeling process is carried out to represent each comment, the modeling process in this study uses the LSTM (Long Short Term Memory) model. The Long Short Tem Memory model in this case uses Sigmoid activation as a classifier, because Sigmoid is able to handle 0-1 probability data. The Loss parameter used is Binary Cross Entropy, because the data that will be the output of classification is binary. Optimizer used is Adam.
   The data sharing for training and testing is 90:10, where training data takes over 90% of the data, 143,613 data and 10% of the testing data, 15,958 data. The whole process of training data using the LSTM model will be saved into a model file, which will later be used as predictive data.

D. Evaluation

To determine the performance of the classification model, we need a method for evaluating. In this study the evaluation method is Hamming-Los. In the Hamming Loss method, the calculation is done by means of the total number of misclassifications for the data being tested. The performance seen in Hamming Loss is a representation of the value of Hloss (H). If the smaller the value of H, the accuracy or performance of the classification model that is built the better (Wiraguna et al., 2019). The equation for calculating Hamming Loss is shown in Equation 1.

$$Hloos(H) = \frac{1}{4}\sum_{i=1}^{P}\frac{1}{Q}\,|h(x_i)\Delta Y_i| \tag{1}$$

Where:

P = the amount of data

Q = the number of classes

$h(x_i)\Delta Y_i$ = the number of errors or errors in the classification that occurs

III. RESULTS AND DISCUSSION

The results of the process and stages that have been carried out in the methodology are explained as follows.

Based on the stages of the preprocessing process that uses word embedding successfully extracted a total of 251,092 words that have been carried out before the cleaning process. Furthermore, the words that are extracted are converted into vectors with a maximum number of numbers as attributes in the vector that is equal to 250 attributes. Then the data is divided into training data and testing data with a proportion of 90:10, where the data sharing is shown in Table 1.

Table 1 Proportion Data for Modelling

	The amount of data	Dimension	Number of Classes
Training Data (90%)	143.613	250	6
Testing Data (10%)	15.958	250	6

From the distribution of data will be entered into a model that has been made, where in the study using the LSTM model with activation parameters using Sigmoid, binary crossentropy type of loss, optimizer using Adam. The epoch that was conducted to conduct training data was 5 epochs with a batch size of 64.

The process of each epoch of training data is shown in Table 2.

Table 2 Result of Training Data

Epoch	Loss	Acc	Val-Loss	Val-Acc	Time
1	0.0775	0.9755	0.0523	0.9809	1857s
2	0.0477	0.9825	0.0486	0.9823	1827s
3	0.0437	0.9835	0.0482	0.9822	1677s
4	0.0404	0.9845	0.0482	0.9822	1705s
5	0.0361	0.9858	0.0497	0.9820	1679s

Based on the data shown in Table 2, from the 5 epochs that have been successfully performed it can be seen that the highest Total Accuracy is obtained at the 5th epoch where with an accuracy of 98.58%, while the lowest accuracy is obtained at the 1st epoch with accuracy of 97.55%, this shows that in the first epoch the data have not been trained as a whole so that the last epoch shows better accuracy. In the validation accuracy, namely training data compared to the testing data that has been done before separation, the 2nd epoch shows the best validation accuracy that is equal to 98.23%, while the 5th epoch has a validation accuracy of 98.20%. Based on the time it shows the 5th epoch has the fastest training time when compared to other epoches, which is 1679 seconds.

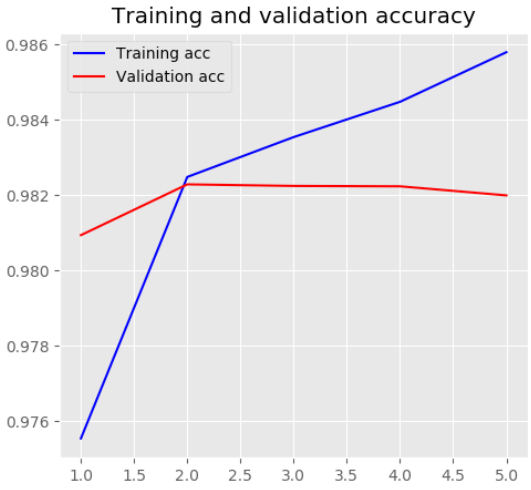

Figure 4 Training & Validataion Accuracy

Based on Figure 4, the X axis represents the number of epochs while the Y axis represents the level of accuracy. When viewed training accuracy continues to increase until the 5th epoch with an accuracy rate of 98.58% while for validation accuracy tends to be stable from the 2nd epoch to the 5th epoch, and has decreased accuracy on the 5th epoch so that the accuracy becomes 98.20%.

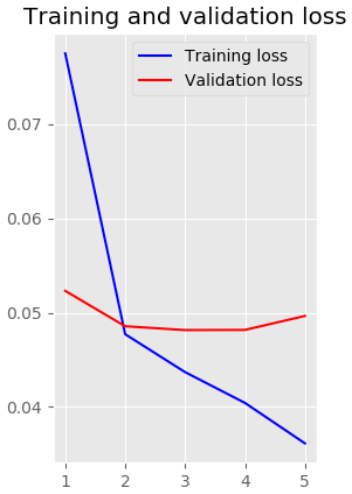

Figure 5 Training & Validataion Loss

Based on Figure 5, the X axis represents the number of epochs while the Y axis represents the loss level. Referring to Figure 5, shows that the level of Loss from training results has decreased significantly, where in the 5th epoch shows a Loss level of 3.6%, this indicates the model created for training has been able to recognize data well. Whereas the validation shows the loss value is stable, this is indicated in the 2nd epoch to the 4th epoch. On the 5th epoch, there was an increase in loss, which was 4.97%.

Overall, the LST model that has been made shows very good performance by successfully increasing the amount of accuracy and reducing the amount of Loss.

In the evaluation of research using the Hamming Loss method, this method calculates the average number of incorrect classifications of testing data, the smaller the value of hamming los shows the excellent performance of the classification model created. The testing data used are data taken from the social media platforms Facebook and Twitter.

Calculation of Hamming loss in this study is to compare testing data with predictive data for each class, so a total of 6 Hamming Loss values are generated. The value of Hamming Loss for each class is shown in Figure 5.

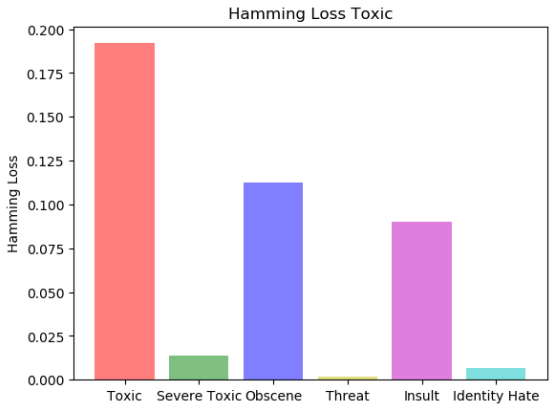

Figure 5 Hamming Loss

In Figure 5 shows the comparison of the results

of Hamming Loss for each class, where the highest value of hamming loss is obtained by the Toxic class, while the lowest value of Hamming Loss is obtained by the Threat class. Details of the Hamming Loss value for each class are shown in Table 3.

Table 3 Result of Hamming Loss

Class	Hamming Loss
Toxic	0.1919968138727116
Sever Toxic	0.013985009532266068
Obscene	0.1122848711185396
Threat	0.0015473609986680944
Insult	0.09031495651719726
Identity Hate	0.0068945705257110025

Based on Table 3 above shows that from among the 6 classes the highest value of Hamming Loss is owned by the Toxic class where the value is 0.19, followed by the Obscene class with a value of 0.112. This indicates that the two classes have a lot of data classified incorrectly. While the lowest value of Hamming Loss is owned by the Threat class with a value of 0.0015 and followed by the Identity Hate class with a value of 0.006. Threat class has the smallest Hamming Loss value when compared to other classes, this shows that the Threat class has the most data classified correctly. If the Hamming Loss value is below 0.001, it is assumed that the classification model created successfully classifies the data class correctly.

IV. CONCLUSION

This research has succeeded in classifying toxic comments on Facebook and Twitter social media for Indonesian texts using the LSTM model. In experiments that have been conducted research conducted as much as 5 epoch training data using the LSTM model, where the comparison of data used for training and testing is equal to 90:10. The validation method used is Hamming Loss successfully calculates the value of the class that has the most classified data which is not good.

The results of experiments conducted using the LSTM model with 5 epochs showed the highest training accuracy obtained at the 5th epoch with an accuracy rate of 98.58%, this shows the performance of the classification model is able to recognize data well. While for accuracy validation is best shown in the 2nd epoch with an accuracy rate of 98.23%, this shows that the classification model created successfully predicts testing data on validation appropriately.

The best loss value resulting from training is obtained at the 5th epoch with a value of 3.6%, the smallest loss value indicates loss of data recognition in a very small model which means the model is able to recognize data well. The best value of validation loss is obtained at epochs 3 & 4 that is equal to 4.82%, this shows the loss of data predictions in the model is very small, which means the model is able to provide all data predictions correctly.

The results of evaluation experiments conducted using Hamming Loss from 6 classes show the Threat class has the smallest Hamming Loss value that is equal to 0.0015, this shows the model is able to classify a lot of data correctly. The smaller value of Hamming Loss shows the accuracy of the model in classifying data correctly.

REFERENCES

Aken, B. Van, Risch, J., Krestel, R., & Alexander, L. (2018). Challenges for Toxic Comment Classification : An In-Depth Error Analysis Challenges for Toxic Comment Classification : An In-Depth Error Analysis, (August). https://doi.org/10.18653/v1/W18-5105

Georgakopoulos, S. V, Vrahatis, A. G., Tasoulis, S. K., & Plagianakos, V. P. (2018). Convolutional Neural Networks for Toxic Comment Classification. *SETN*.

Google, J. (2018). Toxic Comment Classification. Retrieved from https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview

Kallas, P. (2019). Top 15 Most Popular Social Networking Sites and Apps [2019]. Retrieved from https://www.dreamgrow.com/top-15-most-popular-social-networking-sites/

Mestry, S., Bisht, V., Chauhan, R., Tiwari, K., & Singh, H. (2019). MULTI-LABEL CLASSIFICATION OF TOXIC COMMENTS USING FAST-TEXT AND CNN, 18–21.

Ozoh, P., M O, O., & Adigun, A. A. (2019). Identification and Classification of Toxic Comments on Social Media using Machine Learning Techniques. *International Journal of Research and Innovation in Applied Science (IJRIAS)*, (December).

Rumagit, R. Y., & Girsang, A. S. (2018). Prediction Personality Traits of Facebook Users Using Text Mining. *Journal of Theoretical and Applied Information Technology*, *96*(20), 6877–6888.

Rumagit, R. Y., Setiyawati, N., & Bangkalang, D. H. (2019). Comparison of Graph-based and Term Weighting Method for Automatic Summarization of Online News. *4th International Conference on Computer Science and Computational Intelligence 2019 (ICCSCI),* 0–9. https://doi.org/10.1016/j.procs.2019.08.220

Saif, M. A., Medvedev, A. N., & Medvedev, M. A. (2018). Classification of online toxic comments using the logistic regression and neural networks models Classification of Online Toxic Comments Using the Logistic Regression and Neural Networks Models. *AIP Conference Proceedings 2048*, *60011*. https://doi.org/10.1063/1.5082126

Sharma, R., & Patel, M. (2018). Toxic Comment Classification Using Neural Networks and Machine Learning, *5*(9), 47–52. https://doi.org/10.17148/IARJSET.2018.597

Srivastava, S., Khurana, P., & Tewari, V. (2018). Identifying Aggression and Toxicity in Comments using Capsule Network. *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying*, 98–105.

Tala, F. Z. (n.d.). A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia.

Wiraguna, A., Faraby, S. Al, Sc, M., Adiwijaya, P., Si, S., & Si, M. (2019). Klasifikasi Topik Multi Label pada Hadis Bukhari dalam Terjemahan Bahasa Indonesia Menggunakan Random Forest. *E-Proceeding of Engineering*, *6*(1), 2144–2153.