ISSN 2355-0082

Cyberbullying Sentiment Analysis with Word2Vec and One-Against-All Support Vector Machine

Lionel Reinhart Halim[1], Alethea Suryadibrata[2]
[1,2]Department of Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia
[1]lionel.halim@student.umn.ac.id, [2]alethea@umn.ac.id

Accepted on May 19, 2021
Approved on June 09, 2021

***Abstract*—Depression and social anxiety are the two main negative impacts of cyberbullying. Unfortunately, a survey conducted by UNICEF on 3rd September 2019 showed that 1 in 3 young people in 30 countries had been victims of cyberbullying. Sentiment analysis research will be conducted to detect a comment that contains cyberbullying. Dataset of cyberbullying is obtained from the Kaggle website, named, Toxic Comment Classification Challenge. The pre-processing process consists of 4 stages, namely comment generalization (convert text into lowercase and remove punctuation), tokenization, stop words removal, and lemmatization. Word Embedding will be used to conduct sentiment analysis by implementing Word2Vec. After that, One-Against-All (OAA) method with the Support Vector Machine (SVM) model will be used to make predictions in the form of multi labelling. The SVM model will go through a hyperparameter tuning process using Randomized Search CV. Then, evaluation will be carried out using Micro Averaged F1 Score to assess the prediction accuracy and Hamming Loss to assess the numbers of pairs of sample and label that are incorrectly classified. Implementation result of Word2Vec and OAA SVM model provide the best result for the data undergoing the process of pre-processing using comment generalization, tokenization, stop words removal, and lemmatization which is stored into 100 features in Word2Vec model. Micro Averaged F1 and Hamming Loss percentage that is produced by the tuned model is 83,40% and 15,13% respectively.**

***Index Terms*—One-Against-All; multi labelling; sentiment analysis; Toxic Comment Classification Challenge; Word2Vec; word embedding**

I. INTRODUCTION

Cyberbullying refers to bullying that uses electronic technology such as smartphones and the internet. A victim of cyberbullying may increase the risk of low self-esteem [1]. Low self-esteem can cause anxiety and depression [2]. These impacts are supported by the statistics provided by Broadband Search regarding mental health that comes from cyberbullying that depression and social anxiety are in the top 2 ranks [3]. Unfortunately, 1 out of 3 young people in 30 countries has been a victim of cyberbullying [4].

To prevent cyberbullying from happening, detection will be needed. This detection can be achieved by NLP technique which focuses on the interactions between computers and human (natural) languages to do text processing [5]. One of them is sentiment analysis with its ultimate task is to do emotion identification [6]. Sentiment analysis will be used by implementing the Word Embedding approach. This approach will represent words into a vector space and will be achieved by using Word2Vec with Continuous Bag-of-Words (CBoW) model architecture. This model will take words as input and generate vectors as outputs. By using Word2Vec, semantic relationships between words in a sentence can also be found [7]. Thus, Word2Vec has a great role in performing sentiment analysis.

Detection of cyberbullying will be done by using sentiment analysis from Word2Vec and implementing Multi-label Classification. There will be six classes that will be used, namely toxic, severe toxic, obscene, threat, insult, and identity hate. Support Vector Machine (SVM) model will be used to do classification as it is performed better in text processing [8]. Then, One-Against-All (OAA) strategy will be used to be able to implement Multi-label Classification on the SVM.

II. LITERATURE REVIEW

A. *Pre-processing*

Pre-processing is an important step to transform text into a better form with the intention of preparing text for the next step. Pre-processing steps includes [9]:

- Converting all letters to lower case
- Removing stop words
- Removing punctuations
- Converting text into its root forms (lemmatization)

- Splitting the text into smaller pieces (tokenization)

B. *Word Embedding*

Word embeddings are type of word representation in a form of a vector. This approach is widely used in the case of Information Retrieval (IR) and Natural Language Processing (NLP) because of its ability to capture semantic and syntactic information from a word, so that words containing similar meanings can be measured [10].

C. *Word2Vec*

Word2Vec is one of the models used to implement Word Embedding. This model gets input from a collection of texts and generates a vector of the words. This vector can be used to find the proximity of each word in the vector space [11]. Thus, this model can check all the representation that has been learned and displays the closest word [12], as shown in Table I.

TABLE I. WORD COSINE DISTANCE

Word	Distance
spain	0.678515
belgium	0.665923
netherlands	0.652428
italy	0.633130
switzerland	0.622323
luxembourg	0.610033
portugal	0.577154
russia	0.571507
germany	0.563291
catalonia	0.534176

D. *Continuous Bag-of-Words(CBoW)*

CBoW is Word2Vec model rchitectures to create word embedding. The function of this model architecture is to predict a word based on the surrounding words [13]. The network model of CBoW is shown in Fig. 1.

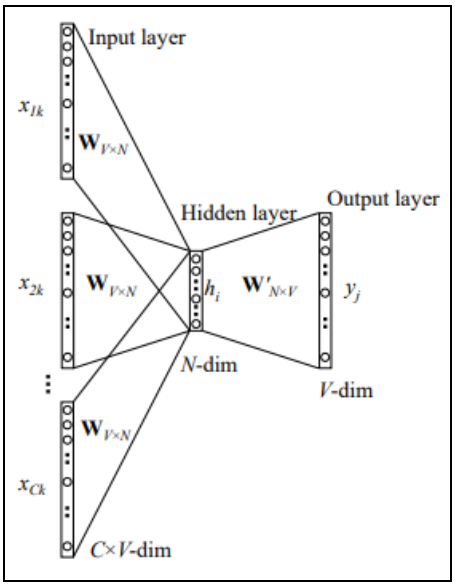

Fig. 1. CBoW model architecture

E. *Support Vector Machine (SVM)*

Support vector machine (SVM) was introduced by Vapnik. The objective of this algorithm is to classify data points by using hyperplane or separator function between classes [14]. There are four hyperparameters used in this algorithm, such as:

- Kernel

  This parameter will affect the type of hyperplane that will be used to separate the data. The Linear kernel will use a linear hyperplane (straight lines as in 2-dimensional space). The Radial Basis Function (RBF) and Polynomial kernels will use a non-linear hyperplane. An illustration of the kernel can be seen in Fig. 2 [15].

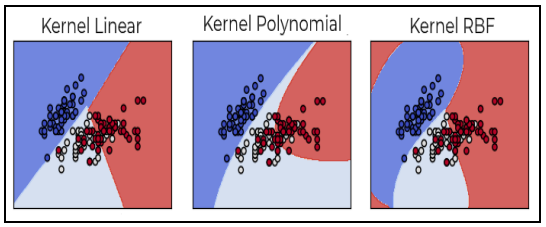

Fig. 2. Illustrations of Linear Kernel (left), Polynomial Kernel (center), RBF Kernel (right)

- Regularization Parameter

  This parameter affects the margin maximization value. The smaller the value, the larger the margin that can be formed. On the other hand, the larger the value, the smaller the margin that will be formed [15].

- Degree

  Degree is a parameter that will affect the flexibility of the hyperplane that is formed. The larger the value, the more flexible the boundary will be [16], as shown in Fig. 3.

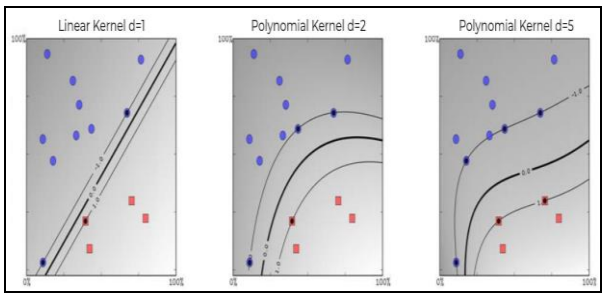

Fig. 3. Illustrations of the differences of degree values

- Gamma

  Gamma determines how big and how far the influence of the training data sample is. If the value is small, then the result is far apart. The result of using different gamma values is shown in Fig. 4 [15].

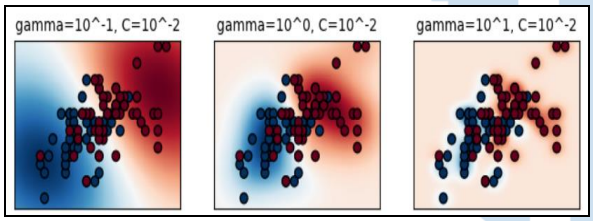

Fig. 4. Illustrations of the differences of gamma values

F. *One-Against-All (OAA)*

One-Against-All is a strategy to train samples to each available class. By doing this, a sample can obtain a binary value for each class and it can be known whether a class is part of the sample or not. OAA has higher accuracy value than One-Against-One and is more suitable for relatively small number of labels [17].

G. *Micro Averaged F1 Score*

Micro Average F1 Score (Micro-f1) is a method to get the average of F1 score values. The Micro-f1 will be calculated as follows [18]:

$$F1_{micro} = \frac{2\,Precision_{micro}\,Recall_{micro}}{Precision_{micro}+Recall_{micro}} \quad (1)$$

The $Precision_{micro}$ is the precision value calculated using micro averaged approach formulated as follows:

$$Precision_{micro} = \frac{\sum_{k\in C} TP_k}{\sum_{k\in C} TP_k+FP_k} \quad (2)$$

And The $Recall_{micro}$ is the recall value calculated using micro averaged approach formulated as follows:

$$Recall_{micro} = \frac{\sum_{k\in C} TP_k}{\sum_{k\in C} TP_k+FN_k} \quad (3)$$

The value of C is the total class that is available. The value of TP_k is the number of true positives in class k. The value of FP_k is the number of false positives in class k. The value of FN_k is the number of false negatives in class k.

H. *Hamming Loss*

Hamming Loss is a metric specifically designed for multi-class (also called multi label) learning [19]. This metric is used to calculate how many misclassified pairs of sample and label. The range of values generated by the Hamming Loss metric is between 0 to 1 or 0 to 100 in percentage.

Smaller value of this metric means better the classification model that has been created. The calculation is carried out using the following equation [20].

$$Hamming\,Loss(h,D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \Delta Z_i|}{|L|} \quad (4)$$

III. METHOD

A. *Dataset*

The dataset that will be used is the Toxic Comment Classification Challenge, available on the Kaggle Page [21]. This dataset is collected from Wikipedia page and has a focus to learn the negative behavior of online chatting. There are around 150.000 records for training data that is provided by this dataset. The dataset is divided into 6 classes, namely toxic, severe toxic, obscene, threat, insult, and identity hate. Fig. 5 shows the first two data in the dataset.

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate	none
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0	1
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0	1

Fig. 5. The first two data in the dataset

B. *System Overview*

The dataset file is in CSV format and will be retrieved in the first process. After retrieving dataset, it will go through pre-processing step. This step includes:

- Generalization, which is the process of converting text into lowercase and removes punctuation.
- Tokenization, which is the process to break a

text into the smallest form without losing its meaning.

- Stopwords Removal, which is the process to omit very common words to give more accurate result, such as 'the', 'a', 'an', 'in', etc.
- Lemmatization, which is the process to change word into its root forms, for example, words 'liked', 'liking', 'and 'likes' will be change to 'like'.

After pre-processing, the dataset is trained to the Word2Vec model. We use CBoW model architecture since this model architecture is faster and considered as the best approach for the use of words that are not unique.

After the process is done, we generalize the data distribution to avoid overfitting. After that, the generalized data will be prepared to be used by SVM. When the data is prepared, it will first pass through Hyperparameter Tuning and the best parameters from this process are used to predict. Finally, the prediction results will be evaluated using Micro Averaged F1 Score and Hamming Loss. Fig. 6 shows the system main flowchart.

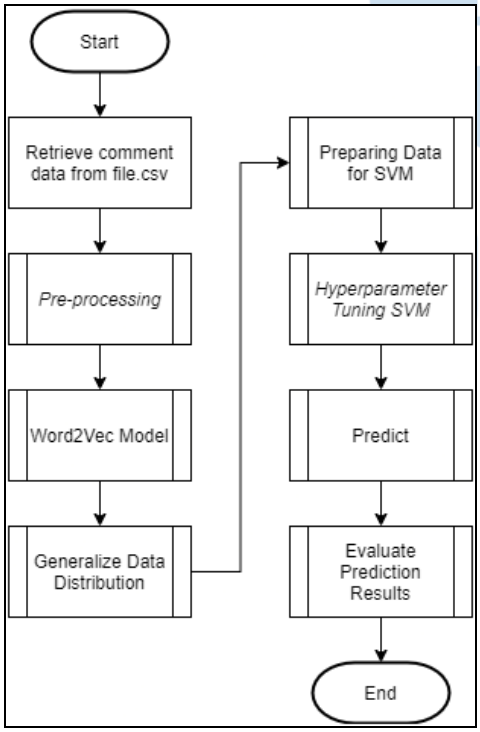

Fig. 6. Main flowchart

IV. Results and Discussion

The experiment is performed with Google Colaboratory and Python version 3. We conducted 4 types of pre-processing as follows.

- Words with stop words
- Words without stop words
- Lemmatized words with stop words
- Lemmatized words without stop words

After pre-processing, each type is trained to the Word2Vec model twice, the first one is using 50 features and the second one is using 100 features. Eight types of Word2Vec models that will be generated are as follows.

- Words with stop words + 50 features
- Words with stop words + 100 features
- Words without stop words + 50 features
- Words without stop words + 100 features
- Lemmatized words with stopwords + 50 features
- Lemmatized words with stopwords + 100 features
- Lemmatized words without stopwords + 50 Features
- Lemmatized words without stopwords + 100 Features

We divided the data so that 70% of the data is for learning and 30% of the data is for testing. Before the training begin, we generalized the data distribution to avoid overfitting. The training data that have been prepared will be used to tune the OAA SVM model. The tuning process will look for the best combination of 3 types of parameters. The three types of parameters are as follows.

- Regularization [0.001, 0.01, 0.1, 1, 10, 100, 1000]
- Kernel ['linear', 'rbf', poly']
- Degree [3,4,5]

After the tuning process, it is found that the best parameter configuration as follows.

- Regularization = 1
- Kernel = RBF
- Degree = 3

Table II shows precision, recall, micro average, micro f1, and Hamming Loss of the model with stop words and 50 features. From 650 testing data, the micro-f1 percentage is 80,77% and Hamming Loss percentage is 17,85%.

TABLE II. Performance of Model with Stop Words + 50 Features

Category	Precision	Recall
Identity Hate	0.74	0.78

Insult	0.75	0.86
Obscene	0.84	0.84
Severe Toxic	0.62	0.79
Threat	0.62	0.81
Toxic	0.95	0.90
Micro Average	0.78	0.84
Micro f1	0.8077	
Hamming Loss	0.1785	

Table III shows precision, recall, micro average, micro f1, and Hamming Loss of the model with stop words and 100 features. From 650 testing data, the micro-f1 percentage is 81,29% and Hamming Loss percentage is 17,05%.

TABLE III. PERFORMANCE OF MODEL WITH STOP WORDS + 100 FEATURES

Category	Precision	Recall
Identity Hate	0.80	0.78
Insult	0.74	0.86
Obscene	0.84	0.81
Severe Toxic	0.64	0.76
Threat	0.66	0.81
Toxic	0.95	0.90
Micro Average	0.80	0.83
Micro f1	0.8129	
Hamming Loss	0.1705	

Table IV shows precision, recall, micro average, micro f1, and Hamming Loss of the model without stop words and 50 features. From 650 testing data, the micro-f1 percentage is 82,23% and Hamming Loss percentage is 16,23%.

TABLE IV. PERFORMANCE OF MODEL WITHOUT STOP WORDS + 50 FEATURES

Category	Precision	Recall
Identity Hate	0.81	0.81
Insult	0.75	0.83
Obscene	0.87	0.86
Severe Toxic	0.66	0.80
Threat	0.64	0.81
Toxic	0.95	0.88
Micro Average	0.80	0.84
Micro f1	0.8223	
Hamming Loss	0.1623	

Table V shows precision, recall, micro average, micro f1, and Hamming Loss of the model without stop words and 100 features. From 650 testing data, the micro-f1 percentage is 82,90% and Hamming Loss percentage is 15,51%.

TABLE V. PERFORMANCE OF MODEL WITHOUT STOP WORDS + 100 WORD2VEC FEATURES

Category	Precision	Recall
Identity Hate	0.81	0.81
Insult	0.75	0.82
Obscene	0.88	0.86
Severe Toxic	0.68	0.80
Threat	0.68	0.82
Toxic	0.95	0.89
Micro Average	0.82	0.84
Micro f1	0.8290	
Hamming Loss	0.1551	

Table VI shows precision, recall, micro average, micro f1, and Hamming Loss of the model with lemmatized words, stop words and 50 features. From 650 testing data, the micro-f1 percentage is 80,73% and Hamming Loss percentage is 17,74%.

TABLE VI. PERFORMANCE OF MODEL WITH LEMMATIZED WORDS + STOP WORDS + 50 FEATURES

Category	Precision	Recall
Identity Hate	0.76	0.76
Insult	0.73	0.85
Obscene	0.82	0.82
Severe Toxic	0.64	0.80
Threat	0.65	0.82
Toxic	0.95	0.89
Micro Average	0.78	0.83
Micro f1	0.8073	
Hamming Loss	0.1774	

Table VII shows precision, recall, micro average, micro f1, and Hamming Loss of the model with lemmatized words, stop words and 100 features. From 650 testing data, the micro-f1 percentage is 81,92% and Hamming Loss percentage is 16,45%.

TABLE VII. PERFORMANCE OF MODEL WITH LEMMATIZED WORDS + STOP WORDS + 100 FEATURES

Category	Precision	Recall
Identity Hate	0.79	0.79
Insult	0.76	0.86
Obscene	0.84	0.82
Severe Toxic	0.65	0.77
Threat	0.69	0.80
Toxic	0.95	0.90
Micro Average	0.80	0.84
Micro f1	0.8192	
Hamming Loss	0.1645	

Table VIII shows precision, recall, micro average, micro f1, and Hamming Loss of the model with lemmatized words, without stop words and 50

features. From 650 testing data, the micro-f1 percentage is 82,29% and Hamming Loss percentage is 16,28%.

TABLE VIII. PERFORMANCE OF MODEL WITH LEMMATIZED WORDS WITHOUT STOP WORDS + 50 FEATURES

Category	Precision	Recall
Identity Hate	0.79	0.79
Insult	0.75	0.84
Obscene	0.86	0.87
Severe Toxic	0.66	0.82
Threat	0.65	0.82
Toxic	0.95	0.89
Micro Average	0.80	0.85
Micro f1	0.8229	
Hamming Loss	0.1628	

Table IX shows precision, recall, micro average, micro f1, and Hamming Loss of the model with lemmatized words, without stop words and 100 features. From 650 testing data, the micro-f1 percentage is 83,40% and Hamming Loss percentage is 15,13%.

TABLE IX. PERFORMANCE OF MODEL WITH LEMMATIZED WORDS WITHOUT STOP WORDS + 100 FEATURES

Category	Precision	Recall
Identity Hate	0.81	0.83
Insult	0.76	0.82
Obscene	0.88	0.86
Severe Toxic	0.68	0.82
Threat	0.67	0.81
Toxic	0.96	0.90
Micro Average	0.82	0.85
Micro f1	0.8340	
Hamming Loss	0.1513	

Table X shows the overall performance for each classifier model. Micro-f1 metric requires high value to be considered as a good model and Hamming Loss metric requires low values to be considered as a good model. Table X also shows that the model which goes through lemmatization, without stop words, and with 100 features is the best model in this experiment. The Micro-f1 percentage is 83,40% and Hamming Loss percentage is 15,13%.

TABLE X. OVERALL PERFORMANCE FOR EACH MODEL

Model	Micro-f1	Hamming Loss
With Stopwords + 50 features	80.77%	17.85%
With Stopwords + 100 features	81.29%	17.05%
Without Stopwords + 50 features	82.23%	16.23%
Without Stopwords + 100 features	82.90%	15.51%
Lemmatize + With Stopwords + 50 features	80.73%	17.74%
Lemmatize + With Stopwords + 100 features	81.92%	16.45%
Lemmatize + Without Stopwords + 50 features	82.29%	16.28%
Lemmatize + Without Stopwords + 100 features	83.40%	15.13%

After getting the values of Micro-f1 and Hamming Loss from each model, we chose the best model for the prediction using OAA SVM. Fig. 7 shows predicting process to evaluate whether the comments "Your brain is now working, you are so idiot!" contain cyberbullying or not.

```
comment = "Your brain is not working, you are so idiot!"
predict(comment)
```

Fig. 7. Predicting process

Fig. 8 shows the result of the pre-processing stage for generalization. In this process, there was a word conversion into lowercase and punctuation removal.

```
Preprocessing Phase 1 - Generalize Sentence:
your brain is not fucking working you are so idiot
```

Fig. 8. Generalization results

The result of the pre-processing stage for stop words removal is shown in Fig. 9. Common words are removed in this process, such as "your", "is", "not", "you", "are", and "so".

```
Preprocessing Phase 2 - Stopwords Removal:
['brain', 'fucking', 'working', 'idiot']
```

Fig. 9. Stop Words Removal Results

Fig. 10 shows the result of the pre-processing stage for lemmatization. Words are changed into its root forms.

```
Preprocessing Phase 3 - Lemmatize:
['brain', 'fuck', 'work', 'idiot']
```

Fig. 10. Lemmatization results

After the pre-processing stage, data preparation is performed to be used by the SVM model as shown in Fig. 11. This preparation will average all words from the context, making it exactly has 100 features.

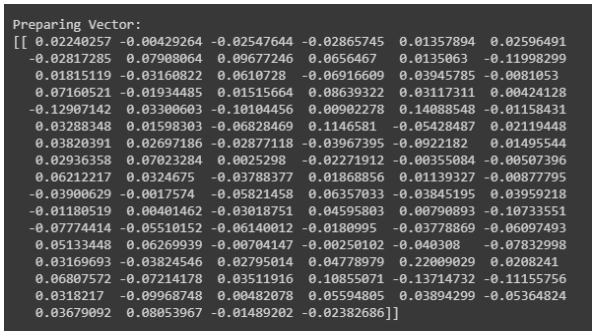

Fig. 11. Averaged vectors

Fig. 12 shows the result of the prediction probability values. This probability can be used if there is a need to create own threshold.

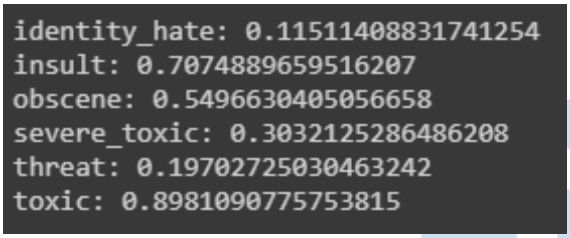

Fig. 12. Prediction probability values

Fig. 13 shows the result of prediction in text value. From this result, it can be concluded that the sentence "Your brain is not working, you are so idiot!" contains cyberbullying in the form of insult, obscene, and toxic.

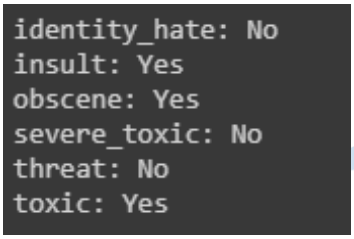

Fig. 13. Prediction result in text

V. Conclusion and Future Work

Based on the research that has been conducted, it can be concluded that the Word2Vec and OAA SVM methods can be implemented to carry out cyberbullying sentiment analysis. The most optimal model based on hyperparameter tuning is by using pre-processed words (lemmatized and without stop words) and 100 features in the Word2Vec model. Then, using Regularization value by 1, RBF Kernel, and Degree value by 3 in the OAA SVM model. Micro Averaged F1 and Hamming Loss percentage that is product by this tuned model is 83,40% and 15,13% respectively.

Since the prediction model that is used is still classifying labels independently, there is no relation between one label with another. The final result still in the form of a model. Therefore, a classifier model that can also determine the relationship between labels like Classifier Chains might be a consideration for future research.

References

[1] J. W. Patchin and S. Hinduja, "Cyberbullying and Self-Esteem". *Journal of school health*. 2013.

[2] J. F. Sowislo and U. orth, "Does low self-esteem predict depression and anxiety? A meta-analysis of longitudinal studies". *Psychological bulletin*. 2013.

[3] 51 Critical Cyber Bullying Statistics in 2020, 2020, available at broadbandsearch.net/blog/cyber-bullying-statistics.

[4] UNICEF, UNICEF poll: More than a third of young people in 30 countries report being a victim of online bullying, 2019, available at https://www.unicef.org/press-releases/unicef-poll-more-third-young-people-30-countries-report-being-victim-online-bullying.

[5] O. Oueslati, A. I. S. Khalil and H. Ounelli, "Sentiment Analysis for Helpful Review Prediction". *International Journal of Advanced Trends in Computer Science and Engineering* Vol.7, No.3, 2018.

[6] A. Rusli, A. Suryadibrata, S. B. Nusantara and J. C. Young, "A Comparison of Traditional Machine Learning Approaches for Supervised Feedback Classification in Bahasa Indonesia". *IJNMT (International Journal of New Media Technology)* Vol.7, No.1, PP.28-32, 2020.

[7] J. C. Young and A. Rusli, "Review and Visualization of Facebook's FastText Pretrained Word Vector Model". *International Conference on Engineering, Science, and Industrial Applications (ICESI)* PP.1-6, 2019.

[8] P. P. Sihombing, R. Jayadim E. Chandra and S. Liu, "Support Vector Machine-Based Hoax Detection on Indonesian Online News". *International Journal of Advanced Trends in Computer Science and Engineering* Vol.9, No.4, 2020.

[9] A. Esuli and F. Sebastiani, "Training Data Cleaning for Text Classification". *Conference on the Theory of Information Retrieval* PP.29-41, 2009.

[10] Y. Liu, Z. Liu, T. S. Chua, M. Sun. "Topical Word Embeddings". *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.

[11] S. Kadam, A. Gala, & P. Gehlot. Word Embedding Based Multinomial Naive Bayes Algorithm for Spam Filtering. *2018 Fourth International Conference on Computing Communication Control and Automation*, 1-5, 2018.

[12] Google, Google Code Archive – Long-term Storage for Google Code Project Hosting, 2013, available at https://code.google.com/archive/p/word2vec/

[13] R. Zhao, K. Mao. "Fuzzy Bag-of-Words Model for Document Representation". *IEEE Transactions on Fuzzy Systems*, 794-804. 2017.

[14] C. Silva and B. Ribeiro, "The Importance of Stop Word Removal on Recall Values in Text Categorization". *Proceedings of the International Joint Conference on Neural Networks* PP.1661-1666, 2003.

[15] F. Pedregosa, et al. "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research*, 2825-2830 vol. 12. 2011.

[16] A. Ben-Hur, C. S. Ong, S. Sonnenburg, B. Schölkopf and G. Rätsch. "Support Vector Machines and Kernels for Computational Biology". *PLoS Computational Biology*. 2008.

[17] K. Milgram, M. Cheriet and R. Sabourin, "One Against One or One Against All: Which One is Better for Handwriting Recognition with SVMs?". 2006.

[18] V. V. Asch, "Macro and Micro-averaged Evaluation Measures". *Belgium: CLiPS* Vol.49, 2013

[19] M. Zhang and K. Zhang, "Multi-label Learning by Exploiting Label Dependency". Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining PP.999-1008, 2010.

[20] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An Ensemble Method for Multilabel Classification". *European Conference on Machine Learning* PP.406-417, 2007

[21] Kaggle, Toxic Comment Classification Challenge. Kaggle, 2018, available at https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview/timeline.