Ann PUBLISHER
Journal of Engineering Sciences (Improsci)
e-ISSN : 3031-7088
p-ISSN : 3032-3452

Utilizing Translation to Enhance NLP Models in Offensive Language and Hate Speech Identification

Sandy Kurniawan[1], **Indra Budi**[2]
[1]Universitas Diponegoro, Semarang, Indonesia, sandy@live.undip.ac.id
[2]Universitas Indonesia, Depok, Indonesia, indra@cs.ui.ac.id

Abstract. The number of social media users in Indonesia has increased in recent years. The surge in social media users leads to more offensive language on these platforms. The use of offensive language can trigger conflicts between users. Therefore, it is necessary to identify the use of offensive language on social media. This study focused on identifying offensive language, hate speech, and hate speech targets on Twitter. The data used were obtained from previous research on identifying offensive language and hate speech. The amount of data is very influential on the performance of the classification. Therefore, data was added using translation in this study. Classical machine learning (SVM et al.) and deep learning (BiLSTM, CNN, and LSTM) algorithms are used as classification algorithms with word n-gram and word embedding as the features. Three scenarios were done based on the training data used in the classification model development. The result shows that scenario 3, which uses translation for data augmentation, can improve the classification model's performance by 5%.

Keywords: Deep Learning, Hate Speech, Offensive Language, Text Classification, Twitter

INTRODUCTION

The number of active social media users in Indonesia has recently increased significantly. This is shown from the survey results conducted by DataReportal in 2020 and 2021, which showed an increase in users by 10 million from the previous year (DataReportal, 2020, 2021). The number of active social media users in 2021 is equivalent to 61.8% of the entire population in Indonesia in January 2021. The increase in social media users has also led to an increase in offensive language. This can be seen from the number of cases handled by the Directorate of Cybercrime Bareskrim Polri (Indonesian Police) in 2018 and 2019. The cases of hate speech handled by the Polri in 2018 were 255 (Arnaz, 2019). This number increased in 2019 when there were 675 cases of hate speech handled by the Polri (Anhari, 2019). The freedom to express oneself on social media is one of the reasons for the emergence of offensive language in the content created by users on social media (MacAvaney et al., 2019). The use of offensive language in social media is a serious problem. Offensive language aimed at a specific target, which is hate speech, can cause emotional instability and affect the mental health of social media

users (Mohaouchane et al., 2019). Therefore, an automatic identification mechanism is needed to prevent the dissemination of offensive language content.

Any form of communication that aims to anger one or more individuals can be considered offensive language. In this case, the form of communication can be hate speech, profanity, bullying, and harassment (Pelle et al., 2018). An offensive language used to mock or insult somebody or a group of people can be categorized into several types, such as taunts, slurs, racism, and extremism (Razavi et al., 2010). On the other hand, according to Komnas HAM (National Commission of Human Rights) (2015), hate speech refers to actions rooted in animosity, whether expressed directly or indirectly, targeting individuals or groups through various methods. The means used to spread hate speech are not limited to electronic media. Komnas HAM states that hate speech can be conveyed through campaigns, banners, religious lectures, and print media. Offensive language and hate speech have a fundamental difference. The difference is in the target of the speech delivered. Offensive language does not always have a specific target, whereas according to Komnas HAM, hate speech has a specific target, such as individuals or groups. An example of offensive language that does not have a specific target is offensive language in the category of dirty words.

Offensive language and hate speech identification can be done using machine learning. One of the challenges that arise when using machine learning is the amount of data required. When used in classification, there is a difference in performance between classical machine learning and deep learning. Based on the data used, the performance of classical machine learning will improve but tend to stagnate at a certain point, while for deep learning, the performance increases as the data used increases (Alom et al., 2019). Therefore, much data is needed to produce a good classification model. However, the amount of data related to offensive language and hate speech in Indonesian is still tiny. Therefore, additional data needs to be done. Data augmentation can be used to address this problem. Data augmentation of text data in the form of synonym replacement, random insertion, random swap, and random deletion was done by Wei and Zou (2019). The study showed that the augmentation techniques used slightly improved the performance. In addition, Sennrich et al. (2016) proposed to use data translation as a data augmentation technique. Unlike previous studies, data augmentation using translation techniques improved performance. Therefore, we performed the data augmentation using the translation technique.

This study focuses on conducting offensive language, hate speech, and hate speech target identification. The data used were obtained from previous studies provided by Ibrohim and Budi

(2019) and Zampieri et al. (2019). In order to increase the amount of data used, translation was used as data augmentation to create synthetic data. The classification problem is solved using a multilabel classification approach, while the algorithms used are classical machine learning and deep learning. The article is structured as follows: Section 2 reviews related literature. Section 3 describes the methods. Section 4 discusses the results and analysis. The final section concludes and suggests future research directions.

LITERATURE

Many studies on offensive language and hate speech have been done in recent years. Pelle et al. (2018) proposed an ensemble classification model to solve the problem of hate speech identification in Twitter and news data. It consists of three classifiers based on word2vec, doc2vec, and SVM. They compared the ensemble model with individual base classifiers, and the ensemble model outperformed all individual base classifiers. In Nikolov and Radivchev (2019), BERT, proposed by Devlin et al. (2019), was proposed to solve the problem of offensive language identification, offense type categorization, and offense target identification. The BERT-based model outperforms standard models on the first and third problems but not on the second problem.

As for Indonesians, the study conducted by Alfina et al. (2017) is a preliminary study on hate speech. This study only classified hate speech into two classes. Furthermore, this study built a Twitter hate speech dataset in Indonesian, consisting of 260 tweets for each hate speech and non-hate speech class. They classified hate speech using classical machine learning algorithms and presented them as the preliminary benchmark for hate speech classification in Indonesia. On the other hand, the study conducted by Ibrahim and Budi (2018) is considered the preliminary study for offensive language identification in Indonesian. Like Alfina et al. (2017), this study also built a Twitter dataset for offensive language identification in Indonesian, consisting of 2.016 tweets. They also provided typo and slang word dictionaries, which can be used for text normalization. In their experiment for offensive language identification, they used various classical machine learning algorithms and n-gram models. The result showed that Naïve Bayes algorithms outperformed other algorithms.

The study by Ibrahim and Budi (2019) proposed a multilabel hate speech and abusive language detection for Twitter in Indonesian. They used problem transformation methods such as Binary Relevance, Label Powerset, Classifier Chain, and classical machine learning algorithms to solve the multilabel classification problem. Furthermore, this study developed the pre-existing

hate speech and offensive language dataset in Indonesian using the dataset from previous studies (Alfina et al., 2017; Ibrohim & Budi, 2018; Putri, 2018). The new dataset consists of 13.169 tweets labeled abusive language, hate speech, hate speech target, category, and level. In another study, Ibrohim et al. (2019) used a deep learning algorithm for abusive language detection using a dataset provided by Ibrohim and Budi (2018). They used LSTM combined with Word2Vec and fastText to classify abusive language. Their proposed method significantly improves the F1-score compared to Ibrohim and Budi (2018). Kurniawan and Budi's (2020) study proposed a classical machine learning approach in hate speech target identification for Twitter. The target labels used are individual and group. Their best performance obtained an F1-score value of 84.77% using SVM as the classification algorithm.

METHOD

This section described the methodology applied in this study. The methodology included data collection, data preprocessing and feature extraction.

Data Collection

The data were collected from a previous study on offensive language and hate speech on Twitter. Two datasets are used in this study. The first is provided by Ibrohim and Budi (2019). This dataset combines the dataset related to offensive language and hate speech in Indonesia from previous studies (Alfina et al., 2017; Ibrohim & Budi, 2018), which is further developed to increase the data. The dataset is in Indonesian and consists of 12 labels such as the offensive label, hate speech label, hate speech target, hate speech category, and hate speech level. This study focuses only on offensive language, hate speech, and hate speech target labels; thus, we do not use the other labels. This dataset consists of 13.169 tweets. The amount of each label is shown in Table 1.

Table 1 The First Dataset Label Detail

Offensive	Hate Speech	Individual	Group	Total
0	0	0	0	5.860
0	1	0	1	885
0	1	1	0	1.381
1	0	0	0	1.748
1	1	0	1	1.101
1	1	1	0	2.194
Total				13.169

Source: Ibrohim and Budi (2019)

The second dataset was provided by the study by Zampieri et al. (2019) called OLID (Offensive et al. Dataset). The second dataset is in English and consists of three labels: offensive language identification, offense type, and offense target. Since the labels are not similar to the first dataset, with the second dataset having no hate speech label, we adjusted the second dataset. We follow the definition of hate speech provided by Komnas HAM to define the hate speech label for the label adjustment. The label adjustment is shown in Table 2. The second dataset consists of 13.240 tweets, and the amount of each label is shown in Table 3.

Table 2 The Second Dataset Label Adjustment

Original Label	Adjusted Label
OFF	Offensive
NOT	Not Offensive
TIN	Hate Speech
UNT	Not Hate Speech
IND	Individual
GRP	Group
OTH	Group

Source: Research Data

Table 3 The Second Dataset Label Detail

Offensive	Hate Speech	Individual	Group	Total
0	0	0	0	8.840
1	0	0	0	524
1	1	0	1	1.469
1	1	1	0	2.407
Total				13.210

Source: Zampieri et al. (2019)

Data Preprocessing

After the dataset is collected and the labels are adjusted, then the dataset is preprocessed to remove unnecessary features or noise. Several processes are included in data preprocessing. Those processes include data translation, emoji removal, case folding, special character removal, stopwords removal, and stemming.

a) Data translation: Data translation is applied only to the second dataset since the primary language used for the classification is Indonesian. The translation process is done using python library, which uses Google Translate API.

b) Emoji removal: Each emoji in the dataset is removed in this step. The reason behind this step is that the emoji has a different representation in both datasets. The emoji removal is done using python library called tweet-preprocessor.

c) Case folding: In this step, all strings in the data are converted into the exact representation for each data. For this study, all strings are converted into lowercase.
d) Special character removal: In this step, all special characters that frequently appeared in tweets are removed. The special characters in tweets included number, punctuation, retweet, and hashtag. Those special characters are considered unnecessary in the classification process.
e) Stopwords removal: Common words that do not provide any helpful information for classification are called stopwords. These words are then removed in this step. The stopwords list for Indonesian used in this study was obtained from a study conducted by Tala (2003).
f) Stemming: The stemming process removes various affixes in each word. This process can reduce the number of tokens obtained. The reduction in the number of tokens can speed up the computational time for classification. Stemming for Indonesian in this study is implemented using Sastrawi Stemmer.

Feature Extraction

After the dataset is preprocessed, the next step extracts the information in the data through the features. These features were used as the input for the algorithms in the classification model development process. In this study, we used the n-gram model as the feature for the classical machine learning algorithms. The n-gram model feature used in this study is the word-level n-gram model. The number of n-grams we used are unigram, bigram, and trigram.

On the other hand, for the deep learning algorithms, we implement word embedding. Word embedding was used to represent the feature for deep learning algorithms. Using word embedding, the features are represented in dense feature vectors with smaller dimension sizes, making the computational cost more efficient. Two types of word embedding are used for comparison in this study. The first is to build the word embedding based on the training dataset, and the second is to use pre-trained word embedding. For the pre-trained word embedding, we used fastText pre-trained word embedding, which was built on Common Crawl and Wikipedia dataset (Bojanowski et al., 2017; Grave et al., 2019).

Classification Model

The following process after feature extraction is classification model development. In this study, we used two approaches to develop the classification model. In the first approach, we used classical machine learning algorithms. The algorithms used are Support Vector Machine (SVM),

Logistic Regression (LR), and Random Forest Decision Tree (RFDT). Since those algorithms are meant for binary classification, we used Label Power-set as the problem transformation method. Label Power-set is proven to perform better than other problem transformation methods (Wei & Zou, 2019). We used deep learning algorithms for the second approach in classification model development. The deep learning algorithms we used are Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Bidirectional LSTM (BiLSTM). The dataset was split into training data and testing data in this process. The split ratio was 90% for training data and 10% for testing data.

DISCUSSION

This section described the experiments scenario, the results obtained, and the analysis of the results in this study.

Experiment Scenarios

This study focuses on using translation for data augmentation in offensive language, hate speech, and target identification in Indonesian tweets. The experiment scenario revolved around the dataset used as training data in the classification model development process. Using two approaches, classical machine learning, and deep learning algorithms, we implemented three experiment scenarios based on the training data used. For the first scenario, we build the classification model only using the first dataset. The first dataset is originally in Indonesian. This scenario was carried out to determine the baseline performance for classical machine learning and deep learning algorithms. In the second scenario, we only used the second dataset, translated from English to Indonesian, to build the classification model. This scenario was carried out to determine the performance of translated data as training data in classification.

Furthermore, the result was analyzed to determine the impact of translation on the classification performance compared to the first scenario. The third scenario combined the first and second datasets to build the classification model. This scenario was conducted to determine the effect of translation as data augmentation and increasing the amount of training data using translation in offensive language, hate speech, and target identification in Indonesian tweets. Each dataset was split into two parts, and then each part was combined. There were three combinations of training data, the details of which are shown in Table 4.

Table 4 Dataset combinations for the third scenario

Combination	First Dataset		Second Dataset		Total
	50%	50%	50%	50%	
A	✓		✓		11.884
B		✓		✓	11.884
C	✓	✓	✓	✓	23.768

Source: Research Data

The experiment scenarios were evaluated using the F1-score. F1-score is the harmonic mean between precision and recall. The results for each scenario are shown in Table 5 and Table 6. Table 5 shows the classification results using classical machine learning algorithms, while Table 6 shows the classification results using deep learning algorithms. As shown in Table 5, the results for the first scenario using classical machine learning achieved the best F1-score of 57.57% using SVM, followed by LR and RFDT. Each used the word unigram as the feature representation. The best feature of the classical machine learning algorithm was the word unigram, as word unigram outperformed the word bigram and trigram on every algorithm. As for the algorithm, SVM outperformed every other algorithm on each n-gram model. Like the first scenario, SVM outperformed LR and RFDT on every feature, and word unigram also achieved the highest performance for each algorithm. The best performance was achieved by SVM, followed by LR and RFDT. Three data training combinations are used in the third scenario. The experiment scenario for combination C, which used RFDT and word trigram, could not be carried out due to the RAM limitation problem. The result shows similar performance with the first and second scenarios, where SVM outperformed the other two algorithms. On the other hand, word unigram became the best feature since it obtained the best performance for each algorithm used on every combination. Regarding the data training combination used in this scenario, the highest performance for combinations A, B, and C obtained an F1-score with a slight difference, with combination C obtaining the highest F1-score.

Table 5 Results for classical machine learning algorithms

Scenario	Algorithm	F1-score (%)		
		Word Unigram	Word Bigram	Word Trigram
1	LR	54.60	33.31	11.66
	RFDT	53.09	36.23	20.12
	SVM	57.57	44.86	25.66
2	LR	28.76	10.10	2.21
	RFDT	25.29	13.95	4.16
	SVM	36.73	20.01	7.92
3A	LR	56.80	23.44	4.87
	RFDT	58.12	32.33	12.91
	SVM	61.18	42.41	18.65

3B	LR	56.02	23.28	4.13
	RFDT	56.49	33.23	11.42
	SVM	61.60	43.07	19.83
3C	LR	61.16	31.68	9.85
	RFDT	59.81	39.36	-
	SVM	63.20	47.91	26.14

Source: Research Data

The result of the classification using deep learning algorithms is shown in Table 6. The highest F1-score in the first scenario was obtained using BiLSTM with training word embedding from the data. The result showed that training word embedding from the data performed better than pre-trained word embedding since BiLSTM and LSTM gave better F1 scores than pre-trained word embedding. The second scenario result using deep learning algorithms showed that pre-trained word embedding performed better than training word embedding on the data. On every algorithm, the best performance was obtained using pre-trained word embedding. CNN obtained the highest performance of the deep learning algorithms. The use of pre-trained word embedding, which performs better than training word embedding on the data, is assumed because the translated data has similar characteristics to the data used for training the pre-trained word embedding. The highest F1 score in the third scenario was obtained using LSTM and training word embedding on combination C. As for combinations A and B, the highest F1-score was obtained using BiLSTM and CNN, respectively, with training word embedding. There was a difference in the amount of training data used between combinations A, B, and C. Combinations A and B have the same amount of training data, 11.884, while combination C used 23.768. The F1-score in combination C is higher because the amount of training data influences it used compared to combinations A and B. This shows that the more training data used, the better the classification performance.

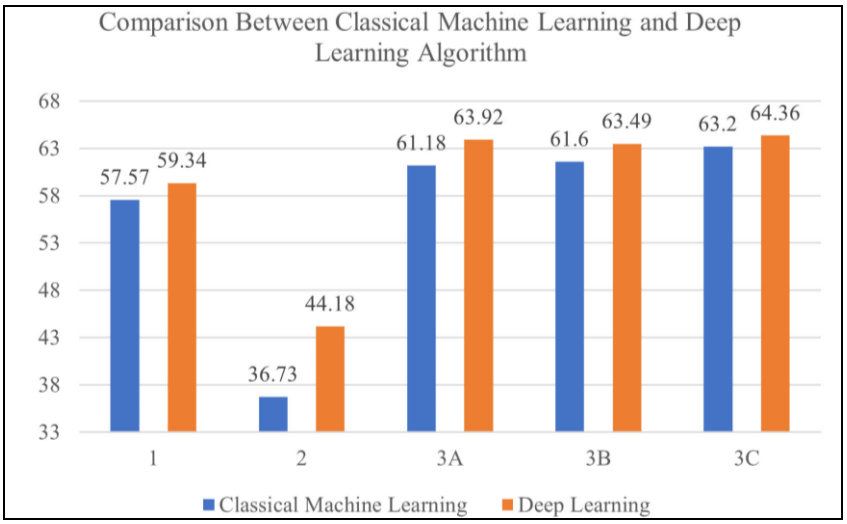
Table 6 Results for deep learning algorithms

Scenario	Algorithm	F1-score (%)	
		Training Word Embedding	Pre-trained Word Embedding
1	BiLSTM	59.34	58.53
	CNN	58.26	58.93
	LSTM	59.11	57.47
2	BiLSTM	29.44	43.21
	CNN	30.98	44.18
	LSTM	34.61	41.36
3A	BiLSTM	63.92	62.63
	CNN	61.08	58.48
	LSTM	60.18	59.89
3B	BiLSTM	61.00	59.13
	CNN	63.49	57.51

	LSTM	52.33	61.63
3C	BiLSTM	63.94	60.71
	CNN	62.75	60.32
	LSTM	64.36	62.15

Source: Research Data

This study used two approaches in offensive language, hate speech and hate speech target identification for Indonesian Twitter. The approaches are classical machine learning algorithms and deep learning algorithms. We used SVM, LR, and RFDT as the classification algorithms in classical machine learning algorithms. The features used are unigram, bigram, and trigram. Label Power-set was used as the problem transformation method to solve the multilabel classification problem in this study. We used CNN, LSTM, and BiLSTM combined with word embedding for deep learning algorithms. We implement two kinds of word embedding applications; the first is training word embedding on the training data, and the second uses pre-trained word embedding provided by fastText. In order to determine which approach is better, we present the best result for both approaches in each scenario and present in Figure 1.



Source: Research Data

Figure 1 Results comparison between classical machine learning and deep learning algorithms

In every scenario, the classical machine learning algorithms obtained the highest F1-score using SVM and word unigram. In contrast, no dominant algorithm exists in the deep learning approach for each scenario. The LSTM algorithm obtained the best performance on scenario 3C and the highest F1-score of all scenarios. On the other hand, CNN became the best algorithm in scenarios 2 and 3B, and BiLSTM became the best algorithm in scenarios one and 3A. Pre-

trained word embedding yielded a poor result regarding the word embedding technique. This is possible because, in the pre-trained word embedding model, the data used to build the word embedding is with formal and standard forms. Since the data used in this study is Twitter data, most of which are non-standard forms such as slang, typos, and many abbreviations, the pre-trained word embedding could have given more results. On the other hand, training word embedding on the data gave better results because the word embedding model was adjusted to the training data. Overall, from Figure 1, we can see that the deep learning approach outperformed the classical machine learning approach in every scenario. Thus, we can conclude that deep learning algorithms are better than classical machine learning algorithms in offensive language, hate speech, and hate speech target identification in Indonesian Twitter.

Regarding the training dataset used in the classification model development as the baseline performance, using the first dataset in the first scenario obtained the best F1-score of 59.34%. The translated data from the second dataset was used in scenario 2. The result of the second scenario shows that the F1-score is less than the first scenario, which is 44.18%. This result shows that the translated data alone is not suitable to be used for offensive language, hate speech, and hate speech target identification. The third scenario was done to determine the effect of combining the first and second datasets. In scenarios 3A and 3B, the amount of data used is 11.884, while scenario 3C used 23.768 data was used in classification model development. The results in the third scenario showed an increase in the F1-score value compared to the first and second scenarios. The difference in the F1-score values is slight, indicating that the distribution of data proportions gives similar results. The highest F1-score value in the third scenario was obtained in scenario 3C, with more training data used than in scenarios 3A and 3B. The results show that using translated data as training data improved the classification performance. This is shown in scenarios 3A and 3B, which have improved compared to the first and second scenarios. In addition, the increase in the training data used in the model development also improved the classification performance. This is shown in scenario 3C, which used more training data than in scenarios 3A and 3B.

To better understand the classification results, we analyzed a confusion matrix focusing on the LSTM model with trained word embeddings, which showed the highest performance. The confusion matrix in Table 7 highlights a higher number of false negatives than false positives across all labels, with 288 false negatives for the Offensive label, 245 for Hate Speech, 339 for Individual, and 243 for Group.

Table 7 Confusion matrix for the classification result

Label	True Positive	True Negative	False Positive	False Negative
Offensive	656	1545	152	288
Hate Speech	599	1511	186	345
Individual	259	1921	122	339
Group	103	2253	42	243

Source: Research Data

The false negative errors occur due to several factors. One of them is that the data used in the development process have an imbalance number of labels. Therefore, the classification model tends to classify the tweet into the negative label on each label. In addition to the imbalance data, the misclassification error also occurs due to other factors such as:

a) Certain words

The first dataset was collected during a political event in Indonesia, leading to specific words ('cebong', 'kampret', 'cina', etc.) and hashtags ('#2019GantiPresiden', '#2019PresidenBaru', '#BalikinKeSolo') being tied to offensive language and hate speech, often resulting in misclassification. Misclassified tweet examples are provided in Table 8, tweet ID 1-4.

b) Meaning of the tweet

The analysis revealed that the classification model struggles to discern the actual intent of tweets, such as distinguishing between statements and sarcasm. Consequently, tweets containing potentially offensive words do not always qualify as hate speech or offensive language, leading to incorrect classification. Misclassifications due to this issue are illustrated in Table 8, tweet ID 5-7

Table 8 Misclassified tweet examples

ID	Tweet example	Cause of error
1	Jgn pernah memilih pemimpin penuh kecurangan seperti ini; #BalikinKeSolo; #BalikinKeSolo	Contains hashtags
2	@USER Selamat hari kartini ; ; #2019GANTIPRESIDEN; #2019PRESIDENBARU	Contains hashtags
3	@USER @USER Ini pak lulung masuk golongan Cebong apa Kampret? \nKok lebih dominan melekat ciri2 Kampret. Apa ini pertanda pak USER?	Contains certain words
4	@USER @USER @USER @USER Iya cebong dan Cina komunis yg cinta sama si jamban	Contains certain words
5	24 jam kedepan pengen buta, budek aja gamau percaya apa apa sip!	Statement tweet
6	@USER Aku pawang monyet, aku pawangnya dia monyetnya	Sarcasm tweet
7	@USER Pak. Rumah Gubernur Lahore akan digunakan sebagai museum dan galeri seni. Anda adalah Menteri Informasi dan Kebudayaan. Harap mengambil beberapa tindakan untuk melindungi tempat dari kotoran dan kerusakan yang disebabkan oleh publik kita yang bodoh. URL	Statement tweet

Source: Research Data

c) Translation result

Regarding the use of translated data for classification model development, we found that translation affected the classification result. Table 9 shows the results of tweet translation from English to Indonesian. We found that the translation result was a textual translation from the tweet, thus changing the meaning or intent of the tweets. The example of this tweet is found in tweets ID 1, 2, and 3. The word 'fuck' in tweet ID 1 was translated into the word 'meniduri', which is not an offensive word in Indonesian because its root form is 'tidur' (sleep). Similar to tweet ID 1, the word 'shit' in tweet ID 2 was translated into the word 'peduli', where the word 'peduli' means care in English. In contrast to tweets ID 1 and 2, in tweet ID 3, the phrase 'holy shit' was translated to the word 'sialan' (damn). In addition, we found that some words in English were lost after the translation process to Indonesian. The example of these tweets is found in tweets ID 4 and 5. The word 'fuck' and 'fucking', which are offensive words in English, were lost when the tweets were translated into Indonesian; therefore, the translated tweet was not an offensive tweet.

Table 9 Tweet translation examples

ID	Original Tweet	Translated Tweet
1	@USER I hope so. That Tana girl fucks anything lol	@USER Saya harap begitu. Gadis Tana itu meniduri apa pun lol
2	@USER And who gives a shit	@USER Dan siapa yang peduli
3	@USER Holy shit. You better not be drinking milk after eating that. Or else...	@USER Sialan. Sebaiknya Anda tidak minum susu setelah makan itu. Atau ...
4	@USER @USER @USER How the fuck can he detect an attitude through a text bubble?	@USER @USER @USER Bagaimana dia bisa mendeteksi sikap melalui gelembung teks?
5	@USER @USER @USER @USER @USER I'm gunna fucking die oh my god	@USER @USER @USER @USER @USER Aku akan mati, ya Tuhan

Source: Research Data

CONCLUSION

In this study on offensive language and hate speech identification in Indonesian, we used data translation for data augmentation. Using three scenarios, we compared classical machine learning algorithms (SVM et al.) with deep learning algorithms (LSTM, CNN, BiLSTM). In the first scenario, the data used to make the classification model was only the Indonesian dataset. The translated dataset was used to make the classification model in the second scenario. Lastly, in the third scenario, three combinations of training data using the Indonesian and translated datasets were used to make the classification model.

We found that using a translated dataset alone resulted in lower performance. However, combining it with the original Indonesian dataset improved the F1 score by approximately 5%, indicating the effectiveness of data augmentation through translation. However, misclassifications, such as false negatives, were influenced by imbalanced data, specific political-related words, the models' inability to understand the true meanings of tweets, and inaccurate translations. To enhance performance further, we suggest ensuring balanced label distribution in training data, possibly through techniques like SMOTE or MLSMOTE, and exploring various hyperparameters in deep learning models.

BIBLIOGRAPHY

Alfina, I., Mulia, R., Fanany, M. I., & Ekanata, Y. (2017). Hate Speech Detection in the Indonesian Language: A Dataset and Preliminary Study. *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 233–238.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. S., & Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. *Electronics (Switzerland)*, *8*(3), 1–67. https://doi.org/10.3390/electronics8030292

Anhari, I. (2019, June 26). *Sepanjang 2019, Polri Telah Tangani 675 Kasus Ujaran Kebencian*. https://hukum.rmol.id/read/2019/06/26/394015/sepanjang-2019-polri-telah-tangani-675-kasus-ujaran-kebencian

Arnaz, F. (2019). *2019, Polri Catat Kasus Hoax Meningkat Tajam*. Berita Satu. https://www.beritasatu.com/nasional/561294/2019-polri-catat-kasus-hoax-meningkat-tajam

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://doi.org/10.1162/tacl_a_00051

Charte, F., Rivera, A. J., Del Jesus, M. J., & Herrera, F. (2015). MLSMOTE: Approaching imbalanced multilabel learning through synthetic instance generation. *Knowledge-Based Systems*, *89*, 385–397. https://doi.org/10.1016/j.knosys.2015.07.019

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

DataReportal. (2020). *Digital 2020: Indonesia*. https://datareportal.com/reports/digital-2020-indonesia

DataReportal. (2021). *Digital 2021: Indonesia*. https://datareportal.com/reports/digital-2021-indonesia

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL HLT 2019*, *1*, 4171–4186.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2019). Learning Word Vectors for 157 Languages. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 3483–3487.

Ibrohim, M. O., & Budi, I. (2018). A Dataset and Preliminaries Study for Abusive Language Detection in Indonesian Social Media. *3rd International Conference on Computer Science and Computational Intelligence 2018*, 222–229. https://doi.org/10.1016/j.procs.2018.08.169

Ibrohim, M. O., & Budi, I. (2019). Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter. *Proceedings of the Third Workshop on Abusive Language Online*, 46–57. https://doi.org/10.18653/v1/w19-3506

Ibrohim, M. O., Sazany, E., & Budi, I. (2019). Identify abusive and offensive language in indonesian twitter using deep learning approach. *Journal of Physics: Conference Series*, *1196*(1). https://doi.org/10.1088/1742-6596/1196/1/012041

Komnas HAM. (2015). Buku Saku Penanganan Ujaran Kebencian (Hate Speech). In *Komisi Nasional Hak Asasi Manusia*. https://doi.org/10.1017/CBO9781107415324.004

Kurniawan, S., & Budi, I. (2020). Indonesian Tweets Hate Speech Target Classification Using Machine Learning. *2020 5th International Conference on Informatics and Computing, ICIC 2020*, 1–5. https://doi.org/10.1109/ICIC50835.2020.9288515

MacAvaney, S., Yao, H. R., Yang, E., Russell, K., Goharian, N., & Frieder, O. (2019). Hate speech detection: Challenges and solutions. *PLoS ONE*, *14*(8), 1–16. https://doi.org/10.1371/journal.pone.0221152

Mohaouchane, H., Mourhir, A., & Nikolov, N. S. (2019). Detecting Offensive Language on Arabic Social Media Using Deep Learning. *2019 6th International Conference on Social Networks Analysis, Management and Security, SNAMS 2019*, 466–471. https://doi.org/10.1109/SNAMS.2019.8931839

Nikolov, A., & Radivchev, V. (2019). Nikolov-Radivchev at SemEval-2019 Task 6: Offensive Tweet Classification with BERT and Ensembles. *Proceedings of the 13th International Workshop on Semantic Evaluation*, 691–695. https://doi.org/10.18653/v1/s19-2123

Pelle, R., Alcântara, C., & Moreira, V. P. (2018). A Classifier Ensemble for Offensive Text Detection. *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web*, 237–243. https://doi.org/10.1145/3243082.3243111

Putri, T. T. A. (2018). *Analisis dan Deteksi Hate Speech pada Sosial Twitter Berbahasa Indonesia*. Universitas Indonesia.

Razavi, A. H., Inkpen, D., Uritsky, S., & Matwin, S. (2010). Offensive Language Detection Using Multi-level Classification. *Canadian Conference on Artificial Intelligence*, 16–27. https://doi.org/10.1007/978-3-642-13059-5_5

Sennrich, R., Haddow, B., & Birch, A. (2016). Improving Neural Machine Translation Models with Monolingual Data. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, 86–96. https://doi.org/10.18653/v1/p16-1009

Tala, F. Z. (2003). *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*.

Wei, J., & Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical*

Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference, 6382–6388. https://doi.org/10.18653/v1/d19-1670

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). Predicting the Type and Target of Offensive Posts in Social Media. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 1415–1420. https://doi.org/10.18653/v1/n19-1144