International Journal on Advanced Science Engineering Information Technology

Vol.9 (2019) No. 4
ISSN: 2088-5334

Translated vs Non-Translated Method for Multilingual Hate Speech Identification in Twitter

Muhammad Okky Ibrohim[#1], Indra Budi[#2]

[#]*Faculty of Computer Science, Universitas Indonesia, Kampus UI, Depok, 16424, Indonesia*
E-mail: [#1]okkyibrohim@cs.ui.ac.id; [#2]indra@cs.ui.ac.id

***Abstract*— Nowadays social media is often misused to spread hate speech. Spreading hate speech is an act that needs to be handled in a special way because it can undermine or discriminate other people and cause conflict that leading to both material and immaterial losses. There are several challenges in building a hate speech identification system; one of them is identifying hate speech in multilingual scope. In this paper, we adapt and compare two methods in multilingual text classification which are translated (with and without language identification) and non-translated method for multilingual hate speech identification (including Hindi, English, and Indonesian language) using machine learning approach. We use some classification algorithms (classifiers) namely Support Vector Machine (SVM), Naive Bayes (NB), and Random Forest Decision Tree (RFDT) with word n-grams and char n-grams (character n-grams) as feature extraction. Our experiment result shows that the non-translated method gives the best result. However, the use of non-translated method needs to be reconsidered because this method needs more cost for data collection and annotation. Meanwhile, translated without language identification method give a poor result. To address this problem, we combine translated method with monolingual hate speech identification, and the experiment result shows that this approach can increase the multilingual hate speech identification performance compared to translate without language identification. This paper discusses the advantages and disadvantages for all method and the future works to enhance the performance in multilingual hate speech identification.**

***Keywords*— social media; multilingual hate speech identification; machine learning.**

I. INTRODUCTION

Hate speech is an act either directly or indirectly to a person or a group based on a feeling of hatred of something inherent in that person or group [1]. In everyday life, the spread of hate speech is a very dangerous act. This is because hate speech can degrade others, cause harm (both material and immaterial), trigger conflict between groups, even to the point of genocide [1]. One example of the dangerous impact of hate speech is the genocide tragedy against ethnic Tutsi in Rwanda in 1994 [2]. This tragedy occurred because some groups alleged the Tutsi ethnic as the cause of increasing social, economic, and political pressure in Rwanda.

Hate speech can be done and disseminated through various means, one of which is through social media. Certain parties often misuse a large number of social media users as a medium for doing and spreading hate speech [3].

Research on hate speech identification in social media continues to grow in recent years. Waseem and Hovy [3] did research on hate speech identification on English Twitter data. They used Logistic Regression with 10-fold cross-validation technique to classify whether a tweet includes hate speech or not. The features used by them include word n-grams and character n-grams. In addition to these two features, their research also used two additional features which are gender (i.e., the sex of people who write hate speech tweet, consisting of the male, female, and unidentified) and location (i.e., the city name where people write hate speech tweet).

For research on hate speech identification in the Indonesian language, [4] researched hate speech identification on Indonesian Twitter data. They used several machine learning algorithms such as Naive Bayes (NB), Support Vector Machine (SVM), Random Forest Decision Tree (RFDT), and Bayesian Logistic Regression with 10-fold cross-validation technique to classify whether a tweet includes hate speech or not. In their research, the features they used are word n-grams, character n-grams, and sentiment lexicon (positive, negative, and neutral).

There have been other studies on hate speech identification. For example, a research on hate speech identification in Facebook and YouTube comments in Hindi using NB and SVM with Term Frequency-Inverse Document Frequency (TFIDF) and word n-grams features [5]; Italian Facebook comments using Long Short-Term

Memory (LSTM) and SVM with word n-grams, POS tag, and lexicon features [6]; and Dutch social media comments using SVM with racist dictionary feature [7].

We can see that there has been a lot of research on hate speech identification in various social media indifferent language using various approaches [8]–[10]. However, there has not been much research on multilingual hate speech identification even though the research on multilingual hate speech identification is needed because many netizens between countries are arguing along with saying hate speech on social media.

In this paper, we adapt and compare two main methods in multilingual text classification that are translated and non-translated sentences for multilingual hate speech identification in social media. We used several Twitter public dataset consisting of Hindi, English, and Indonesian language from several previous research in hate speech identification. The classifier that we used includes NB, SVM, and RFDT with word n-grams and char n-grams features.

In general, this paper is organized as follows. Section II discusses the background theory for this research, which is text classification using machine learning approach in general and multilingual text classification, explains the dataset and method that we used. Section III presents our experimental results. Lastly, Section IV explains our conclusions and future works for this research.

II. Materials and Methods

This section discusses hate speech definition and example, text classification using machine learning in general, and multilingual text classification as a background for this research.

A. Hate Speech

Freedom of expression in social media are often misused by netizens; one of them is spreading hate speech. Hate speech is an action (either directly or indirectly) based on a certain factor of hatred towards a person or a particular group [1]. The factors that are often used as targets or bases of hatred include religion, ethnicity, race, ethnicity, disability, gender, and sexual orientation.

Hate speech can be in the form of acts of humiliation, defamation, provocation, incitement, and all actions that can have a negative impact a person or a group that is the target of said hate speech [1]. Some of the negative effects of hate speech include discrimination, social conflict, material losses, and human victims, and genocide.

1) Discrimination: Discrimination is an act of differentiation, exclusion, or limitation of an individual or group. The existence of hate speech can result in acts of discrimination against a person or a group that makes the person or group get discrimination from the community which results in a reduction in the recognition, acquisition, and implementation of human rights in various fields of life.

2) Social conflict: Hate speech in the form of incitement to be hostile to individuals or groups can lead to conflict. This conflict can be a conflict between individuals, which then extends into a conflict between groups.

3) Material losses and human victims: Social conflicts due to acts of speech hate that are not immediately dealt with quickly and precisely can lead to anarchic conflicts such as brawls and so on that can cause material losses and the emergence of casualties.

4) Genocide: Speech acts of hatred in the form of excessive incitement can make the labeling and negative stigma of community groups against a group of people who are victims of the hate speech. If this is allowed, public hatred of the victims of incitement can increase and lead to anarchic actions that lead to the genocide of the group.

B. Text Classification using Machine Learning Approach

Text classification is a process of placing text data objects into a particular category. In general, text classification steps include data collection and annotation, pre-processing data, features extraction, classification, and evaluation.

Some social media such as Twitter[1] provides Application Programming Interfaces (API) that allows developers and researches to collect public tweets dataset by crawling them. The data has been collected and then annotated by the annotator, both derived from the linguist and crowdsourcing [11]. Besides collecting data by crawling and annotated them, researchers often use a dataset that has been annotated from previous work, so that their research focuses on developing algorithms only.

Before processing the labeled dataset, it is necessary to pre-process the data to streamline the dataset at the time of clarification. In general, pre-process in text classification include tokenization and case folding (document standardization, usually done by lowercase conversion) [12], data cleansing (removing unnecessary character/attribute and punctuation) and token normalization [13], stemming, and stop word removal (dropping common words are not informative) [14].

After pre-processing, the dataset is ready for feature extraction. One of the frequently used features in text classification is word n-grams and character n-grams [3]-[6]. In word n-grams and character n-grams features, each sentence will be regarded as a bag of word/character in the form of a string with length *n* [15]. For example, given a sentences "*he speaks hate speech*", the word 3-grams (word trigrams) will extract this sentences into |*he_speaks_hate*| and |*speaks_hate_speech*|; while character 3-grams (character trigrams will extract this sentences into |*he_*|, |*e_s*|, |*_sp*|, |*spe*|, |*pea*|, |*eak*|, |*aks*|, |*ks_*|, |*s_h*|, |*_ha*|, |*hat*|, |*ate*|, |*te_*|, |*e_s*|, |*_sp*|, |*spe*|, |*pee*|, |*eec*|, and |*ech*|.

The next process is classifying the dataset. Nowadays, many algorithms have been developed to classify different types of data for various purposes. Some classifier (classification algorithms) are often used as baselines in text classification such as NB, SVM, and RFDT [4]-[7].

To evaluate the classification results, there are several techniques that can be used, one of them is *k-fold cross-validation* [16]. In this technique, data will be divided into two parts, i.e. training data and testing data. For example, if we chose $k = 10$, then 9/10 part of data will be used as training data and 1/10 part of data will be used as testing data. The classification process will be repeated k times

[1] https://apps.twitter.com/

(*fold*), where each data will undergo data training and data testing, which is then evaluated based on a particular metric evaluation.

When doing classification, there is something quite important to do, i.e., balancing the number of dataset against each class of data labels. The unbalanced dataset can give negative classification result [17]. This is because the unbalanced dataset between majority and minority class tend to make the classification results in the majority class better than the minority class. The unbalanced dataset problem can be solved by data re-sampling technique. This technique is balancing dataset by duplicating some of the minor class data or deleting some of the major class data such that the dataset on each class become more balance.

C. Multilingual Text Classification

Multilingual text classification is a process to classify text in the multilingual corpus (dataset). In-text processing, often we encounter problems such as the multilingual text classification problem. Some examples of multilingual text classification problems are newspaper categorization [18] and customer feedback mining [19].

In general, the multilingual text classification problem can be solved simply by translated method approach viz. translating all documents in testing data into monolingual training data before classifying it using particular dictionary/translator [20]. Using this technique, we only need monolingual labeled data for the training data to classify a document in many languages. However, this technique has several shortages; one of them is the ambiguity or failure of the translation result. In multilingual text classification, the incorrectly of the translator in translating document can change the classification results. This is because the ambiguity or failure of translation result may give different meaning (semantics) and feature vector that is used to classify the document [20].

Besides the translation method, a multilingual text classification problem can be solved by collecting and combining multilingual labeled document into a training dataset. Next, the other document is classified using that training dataset without the translation process. This method called a non-translated method or language-dependent method because we do not need to translate process and assume the dataset is in the same language. This method can give a high classification result because it does not have translation ambiguity problem, same as doing monolingual text classification. However, this technique requires big labeled dataset from many different languages, that mean this method needs more cost for data collecting and annotation process. For example, if the training dataset just contains document in Hindi, English, and Indonesian language; then we can only classify a document in those three languages.

To bridge the limitation between translated method and non-translated method, we can combine the translated method and monolingual text classification using a language identification approach. This method can increase the classification performance because it has large training dataset in several languages (using the monolingual pre-trained model), while still facilitating documents written in other languages not included in the dataset by translating the document to the main language and classifying it using the main pre-trained model. For example, suppose we have training dataset contain English and Indonesian dataset, and we decide English as the main language. Before classifying a document, the language detector will detect the language of the document first. If the document is written in the Indonesian language, the document will be classified using Indonesian pre-trained model. Otherwise, the document will translate to English and classified using English pre-trained model.

D. Dataset and Method

Twitter is a social media with a huge number of users around the world that often being misused by its users to spread hate speech. In general, the flowchart of the research method in this paper can be seen in Figure 1.

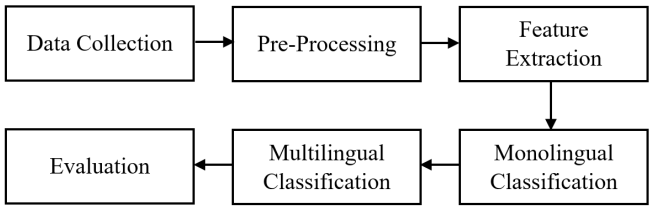

Fig. 1 The flowchart of the research method

First, we collect the Twitter dataset in various languages from some previous researches that open for public. Instead of crawling and annotating the Twitter dataset ourselves, we prefer using the dataset from some previous researches to get more valid ground truth for every dataset language that we used. From our literature review process, we get three languages from some previous researches on hate speech detection that open their dataset for the public, which are Hindi, English, and Indonesian language.

For hate speech dataset in Hindi, we use the dataset of [23]. It is the collection of code-mixed Hindi-English Twitter dataset by scrapping them using Twitter Python API[2] with certain words, phrases, and hashtags about riots, public protests, politics, etc. in Hindi as the queries. Their dataset was annotated by two annotators that have linguistic background and proficiency both in English and Hindi. The dataset was annotated into two labels (hate speech and normal speech) and the final label was decided by 100% agreement strategy. Their annotation process produced 1,661 tweets labeled as hate speech and 2,914 labeled as normal speech.

Next, for hate speech dataset in English, we use a Twitter dataset [9] that collected Twitter dataset in English using Twitter API with English hate words and phrases from https://hatebase.org/ as the queries. Their crawling process collects about 33,458 tweets. From those tweets, they chose 25,297 tweets randomly to be annotated using CrowdFlower workers. 3-6 annotators annotated each tweet in those data into three labels which are hate speech (coded as '0'), offensive (coded as '1'), and neither (non-hate speech nor non-offensive, coded as '2'). The final label in their annotation process was decided using majority voting strategy. In this research, we just use the Twitter dataset of

[2] https://pypi.org/project/twitterscraper/0.2.7/

that labeled as hate speech (contain 1430 tweets) and neither (include 4163 tweets) for our experiment [9].

Meanwhile, for Indonesian dataset, we collect it from three previous types of research [4], [21], [22]. In [4], they are collecting Twitter dataset using Twitter Streaming API with some query that related to the election of Jakarta Governor in 2017 such as *#DebatPilkadaDKI, Pilkada Jakarta 2017, #SidangAhok,* etc. They manually annotated the Twitter dataset into hate speech or non-hate speech using 30 volunteers from the various background in terms of age, gender, ethnicity, and religion to reduce the subjective bias. 3 annotators labeled each tweet, and they used 100% agreement strategy to decide the final label. Their data collection process produces 713 tweet, which has 100% agreement that contains 260 tweets labeled as hate speech and 453 tweets labeled as non-hate speech[3].

Similarly, a study also collected Twitter dataset using the Twitter Streaming API, and each tweet was labeled by 3 annotators and also used a 100% agreement strategy for deciding the final label [21]. Their research is just focused on hate speech against religion, so they do not annotate the Twitter dataset into hate speech or non-hate speech. The dataset in their research annotated into hate speech against religion and non-hate speech against religion. The dataset size from their annotation process contains 900 tweets, where 450 tweets labeled as hate speech against religion and 450 tweets labeled as a non-hate speech against religion. From our study on the dataset [21], although the tweet labeled as a non-hate speech against religion, the tweet can contain hate speech in other categories (whether sexism, slurs, etc.). Thus, we just use the dataset that labeled as hate speech against religion for our research experiment.

The Twitter dataset was annotated just into two labels [4], [21]. The Twitter dataset was annotated into three labels that are non-abusive language [22]. Abusive but not offensive (abusive language in the context of jokes or vulgar conversations), and abusive and hate speech (the abusive language that used to curse someone). They crawled the Twitter dataset using Twitter API and Tweepy Library[4] with abusive words and phrases for the query. They used 20 volunteers to annotate the Twitter dataset, where each tweet was annotated by 3 annotators, and the final label was decided using 100% agreement strategy. Their annotation process collected 2,016 tweets that contain 331 tweets labeled as non-abusive language, 1,090 tweets labeled as abusive but not offensive, and 595 tweets labeled as abusive and hate speech[5].

Before the feature extraction process, we do some preprocessing on our dataset. The data preprocessing that we do in this research consists of case folding, data cleansing, and token normalization. The case folding process is done by changing all characters in our dataset to lower case. Next, we do data cleansing process by removing unnecessary characters such as RT (stand for retweet), username, and Uniform Resource Locator (URL). Lastly, we do token normalization to replace the non-formal words into formal ones. For English hate speech dataset, we normalize the non-formal words using English non-formal words dictionary given by http://luululu.com/tweet/.

Meanwhile, for the Indonesian dataset, we standardize the non-formal words using Indonesian non-formal words dictionary [4][6]. Unfortunately, from our literature review, we do not get non-formal words dictionary for Hindi. Therefore, we do not normalize the Hindi non-formal words in our experiments.

After pre-processing the dataset, we extract several features from our dataset into the features vector. In this research, we used word n-grams and character n-grams features. For word n-grams, we use word unigram, word bigrams, word trigrams, and the combination of word unigram, bigrams, and trigrams. Meanwhile, for character n-grams, we use character trigrams, character quadgrams, and the combination of character trigrams and quadgrams.

To know the best classifier and feature combination for every language dataset, we do monolingual hate speech classification first before multilingual classification. We use three classifiers that are SVM, NB, and RFDT. To validate our classification results, we use 10-fold cross-validation technique [16]. This technique will divide the dataset into a training set (9/10 partition) and testing set (1/10 partition), and the classification process will be repeated ten times (fold) such that every data will become training data and testing data, alternately. For the metric evaluation, we use the F_1-*Score* (usually also called as F_1-*Measure*) as the metric evaluation [24]. The model with the highest F_1-*Score* in each language will be used for the multilingual hate speech identification process. For the multilingual classification process, we use three methods that are and non-translated, translated without language identification, and translated with language identification.

III. Results and Discussions

In this research, we do some experiments in finding the best method and model for multilingual hate speech identification. First, we do monolingual hate speech identification to find the best model for each language. Here, we use SVM, NB, and RFDT with character n-grams and word n-grams to identify whether the tweets are hate speech or not. The monolingual hate speech identification results experiment for each language can be seen in Table I-III. Furthermore, the average F_1-*Score* from every model can be seen in Table IV.

TABLE I
F_1-*Score* for Hindi Dataset (%)

Features	Grams	SVM	NB	RFDT
Character	3	63.30	63.03	61.29
	4	**66.03**	64.10	62.25
	3+4	65.12	64.19	63.60
Word	1	63.48	62.26	58.30
	2	56.00	59.13	57.01
	3	25.44	45.80	33.96
	1+2	65.66	63.62	58.61
	2+3	45.44	58.52	56.75
	1+2+3	64.34	63.53	57.15

[3] https://github.com/ialfina/id-hatespeech-detection
[4] http://www.tweepy.org/
[5] https://github.com/okkyibrohim/id-abusive-language-detection
[6] https://github.com/ialfina/ID-Kamus-Typo

TABLE II
F_1-SCORE FOR ENGLISH DATASET (%)

Features	Grams	SVM	NB	RFDT
Character	3	91.48	89.07	87.22
	4	89.76	89.83	87.87
	3+4	91.53	90.02	88.38
Word	1	**92.36**	89.18	86.55
	2	62.53	62.99	64.59
	3	37.51	27.05	29.13
	1+2	92.23	87.27	83.48
	2+3	57.60	49.19	62.72
	1+2+3	92.19	82.59	82.04

TABLE III
F_1-SCORE FOR INDONESIAN DATASET (%)

Features	Grams	SVM	NB	RFDT
Character	3	80.61	79.43	75.77
	4	81.42	81.46	78.58
	3+4	81.09	80.86	75.83
Word	1	82.25	81.75	79.08
	2	74.70	70.19	68.94
	3	50.56	44.96	62.10
	1+2	82.07	82.03	77.15
	2+3	73.82	68.17	67.12
	1+2+3	81.91	81.73	75.88

TABLE IV
AVERAGE *F_1-SCORE* FOR MONOLINGUAL HATE SPEECH IDENTIFICATION (%)

Features	Grams	SVM	NB	RFDT
Character	3	78.46	77.18	74.76
	4	79.07	78.46	76.23
	3+4	79.25	78.36	75.94
Word	1	79.36	77.73	74.64
	2	64.41	64.1	63.51
	3	37.84	39.27	41.73
	1+2	**79.99**	77.64	73.08
	2+3	58.95	58.63	62.20
	1+2+3	79.48	75.95	71.69

From Table I-III, we can see that SVM with character quadgrams feature is the best model (in our experiment) for hate speech identification in Hindi, while SVM with word unigram feature is the best model for English and Indonesian hate speech identification. This result indicates that every language may have a different best model for text classification, especially in this case, for hate speech identification.

After finding the best model for hate speech identification for every language, we make multilingual hate speech identification. We split every dataset into training data and testing data, and then combine it. Our testing dataset contains 900 tweets, consisting of 300 tweets in Hindi, 300 tweets in English, and 300 in Indonesian. In these multilingual hate speech identification, we experimented with three methods that can be seen in Figure 2.

The first method (namely *non-translated* method) is multilingual hate without translating document (tweet) before classifying it. Here, all training dataset from all language (Hindi, English, and Indonesian) are combined and then trained using SVM with the combination of word unigrams + bigrams (the model was chosen based on the average of *F_1-Score* on monolingual hate speech identification). This pre-trained model is saved as a pickle model. All tweets in the testing dataset are further classified using the pre-trained model.

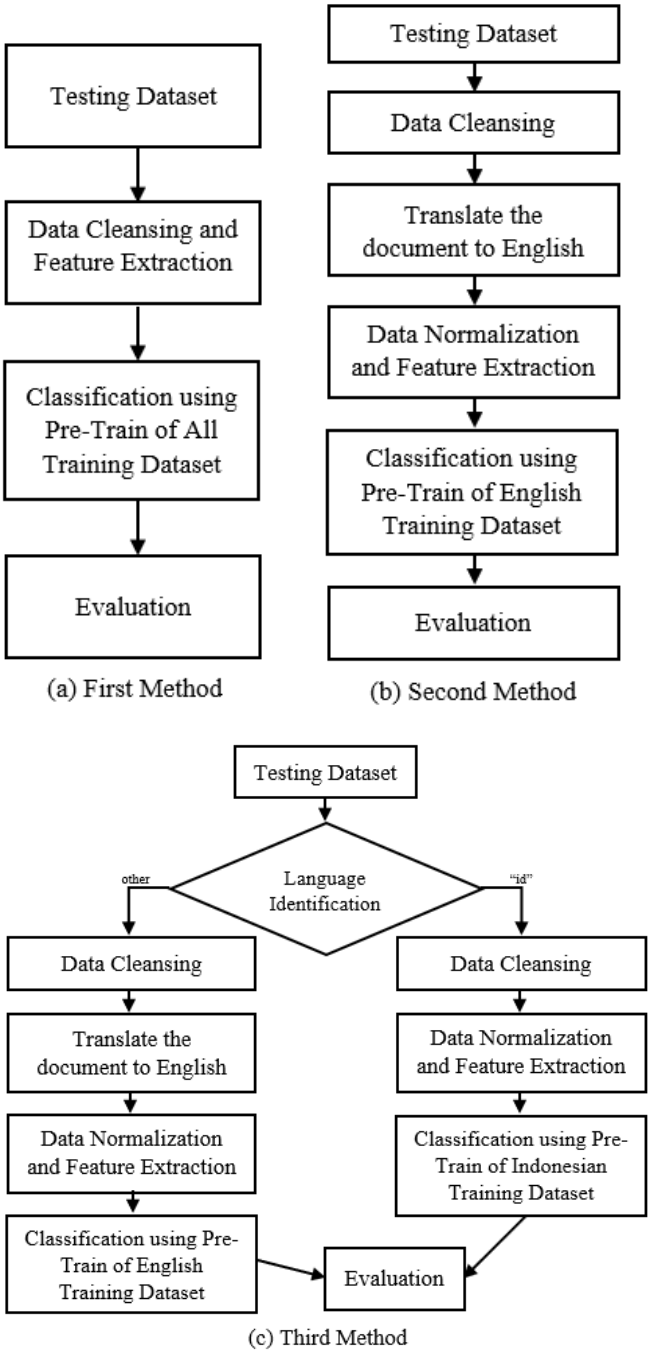

Fig. 2 The flowchart of every method for multilingual hate speech identification experiment

In the second method (namely *translated without language identification method*), we train English training dataset using SVM with word unigram feature and then saved it as a pickle model. Next, all tweets in the testing dataset are translated to English using Google Translate that implemented using Mtranslate Library[7] and then classified using pre-trained English model. We do this scenario to know whether we can transform multilingual hate speech identification into monolingual hate speech identification or not.

[7] https://github.com/mouuff/mtranslate

Besides using standard translated method (which is the translated without language identification method), in the third method, we proposed *translated with language identification method*. This method proposed to know the multilingual hate speech identification performance when we combine the standard translated method with monolingual hate speech identification. We trained the Indonesian training dataset using SVM with word unigram feature and saved it as pickle model for monolingual hate speech identification, while for the translated method we use English pre-trained model was built on the second scenario. Before classifying the tweet on a testing dataset, we identify the language of the tweet using Google Language Detection that implemented using Langdetect Library[8]. If the language of tweet detected as "id" (Indonesian Language) tweet will be classified using Indonesian pre-trained model. Otherwise, the tweet will be translated into English and then classified using pre-trained English model.

Same as in monolingual hate speech identification experiment, we use F_1-*Score* to evaluate the three scenarios that we used in multilingual hate speech identification. The experiment result for multilingual hate speech identification can be seen in Table V.

TABLE V
F_1-SCORE FOR MULTILINGUAL HATE SPEECH IDENTIFICATION (%)

Dataset	F_1-*Score* for each Method		
	Method 1	**Method 2**	**Method 3**
Hindi	70.92	53.33	54.85
English	88.09	74.29	74.23
Indonesian	73.07	47.46	64.74
All	76.95	58.39	65.16

Based on Table V, we can see that the non-translated method gives the best performance for multilingual hate speech identification. This is because the pre-trained model for the non-translated method includes all language that contains in the test set. Here, doing multilingual hate speech identification when all language in test contain in the pre-trained model is same as doing monolingual hate speech classification. However, although it gives the best result, the use of the non-translated method for multilingual hate speech identification needs to be considered because of requires a lot of cost for data annotations.

Meanwhile, our multilingual hate speech identification experiment using translated method has not given such good results. This happens because of several causes. The first cause is the ambiguity of the translation results, where the ambiguity of translation results can change the semantics of text. For example, given Indonesian text "*Goblok lu Anjing*," the Google Translate translates the text into"Stupid dog," instead of "You are a stupid dog." From this example, we can see that the ambiguity of the translation results can change the semantics and give different feature vector of text and moreover can make different classification results. The second cause is the translation failure. In our experiment, Google Translate failed to translate some words, especially words written in other forms (slang forms). For example, Google Translate fails to translate "*goblog*" ("*goblok*", means "stupid/idiot"), such that when we translate "*Goblog lu*", Google Translate gives "*Goblog* you", instead of "You idiot". In the feature extraction process, "*goblog*" from "*goblog* you" will be erased, because of not contained in English pre-trained pickle model vocabulary. Even though "*goblog*" is an abusive word that is often used to convey hate speech on social media [22]. Therefore, the failure of translation also can give wrong classification results. Although the translator provides the true translate of a tweet, the tweets can still be misclassified. This may be caused by the main pre-trained model (English pre-trained model in our case) not cover the hate speech domain from a tweet that will be classified. In our English dataset. The dataset was crawled using English hate speech lexicon that compiled by hatebase.org [9]. This can cause the English dataset that we used do not cover all domain of hate speech because hate speech in various countries can have different topics with different unique hate speech lexicon that translator cannot translate correctly.

Next, from Table V we also can see that the translated with language identification method can give significantly better results compared to translated without language identification method. This indicates that combining translated method with the monolingual hate speech identification can increase the performance in multilingual hate speech identification because in general monolingual text classification gives better results than multilingual text classification. However, the results (F_1-*Score*) are still under 70%. This indicates there are still pretty much misclassifications, included the tweets written in the Indonesian language. Our analysis shows that this caused by the misdetection of the language detector. For example, suppose given an Indonesian tweet that can easily be classified as hate speech using Indonesian pre-trained model; the incorrect language detection result makes the tweet will translate to English and classified using English pre-trained model. As described before, the ambiguity and failure of translation result may cause misclassification.

IV. CONCLUSIONS

Hate speech is a problem that must be taken seriously because it is a very dangerous act. Nowadays, many netizens are typing and posting a hate speech by mixing the language in their social media. This paper has been discussed multilingual hate speech identification using several approaches that are non-translated, translated without language identification, and translated with language identification method. We used hate speech dataset obtained from several previous works containing Hindi, English, and Indonesian hate speech dataset. Before doing multilingual hate speech identification experiment, we do monolingual hate speech experiment to get the best monolingual hate speech identification model for every language.

In this paper, we use several machine learning approaches, namely SVM, NB, and RFDT with simple word n-grams and character n-grams feature. In this paper, we use F_1-*Score* as the metric evaluation in choosing the best model for every language. Our experiment result shows that among the used model, SVM with character quadgrams feature is the best model for Hindi hate speech identification using our dataset. Meanwhile, among the model that we used, the best model for English and Indonesian hate speech identification using

[8] https://pypi.org/project/langdetect/

our dataset is SVM with word unigram feature. For the average, we get SVM with the combination of word unigram and word bigrams as the best model. Here, our experiment results in monolingual hate speech identification also show that different language may have a different best model, by the initial hypothesis.

For multilingual hate speech identification, our experiment shows that the non-translated method gives the best performance. This is because the pre-trained model for the non-translated method includes all language that contains in the test set. However, the use of the non-translated method for multilingual hate speech identification need to be reconsidered. The use of non-translated method needs more dataset that equals need more cost for data annotations process. Especially because of multilingual, we need more effort in searching annotators that native in the language that will be used as the dataset.

On the other hand, the use of translated without language identification method give such poor results in our multilingual hate speech identification experiment. This happens because of several causes such as the ambiguity of translation result, the failure of the translator when translating tweets, and the hate speech domain problem.

By combining the translated method with monolingual classification, our experiment using translated with language identification method shows that this approach can increase the classification performance significantly in multilingual hate speech identification compared to translate without language identification method. This is because in general monolingual text classification gives better results than multilingual text classification. However, the experiment result is still under 70% of F_1-*Score* that indicates there are pretty much tweets that misclassified. This may be caused by the misdetection of the language detector such that tweet that should be classified using Indonesian pre-trained model is even translated to English and then classified using English pre-trained model, where previously mentioned that the ambiguity and failure of translation result might cause misclassification.

For future works, several ways may enhance multilingual hate speech identification performance. The basic way is to try a different approach to finding the best model for every language in monolingual hate speech identification. Future research can use different classifiers and features. Moreover, future works can use deep learning approach such as Long Short-Term Memory (LSTM) with word embedding. Several works in hate speech identification have been shown that this approach gives a good result both in English [25] and Indonesian language [26], [27].

To handle the translator (the ambiguity and failure of translation result) and language detector issue, future works may use and comparing different translator and language detector from several providers such as Microsoft Translator[9], Yandex Translate[10], IBM Watson Language Translator[11], etc.

Last, for hate speech domain issue, future works may use multilingual hate speech terms lexicon as additional features. For this, we can get a collection of hate speech terms lexicon from various countries from hatebase.org. The use of multilingual hate speech terms lexicon as additional features can overcome the failure of the translator when translating a hate speech terms which has been exemplified before. Furthermore, the multilingual hate speech terms also can use as language detector tools. For example, if a tweet contains Indonesian hate speech terms, this tweet can be classified as Indonesian tweet and further can be classified using Indonesian pre-trained model to decide whether the tweet contains hate speech or not.

[9] https://www.microsoft.com/en-us/translator/business/translator-api/
[10] https://tech.yandex.com/translate/
[11] https://www.ibm.com/watson/developercloud/language-translator/api/v2/curl.html?curl##introduction

ACKNOWLEDGMENT

The authors acknowledge the PITTA A research grant NKB-0350/UN2.R3.1/HKP.05.00/2019 from Directorate Research and Community Services, Universitas Indonesia.

REFERENCES

[1] Komnas HAM, *Buku Saku Penanganan Ujaran Kebencian (Hate Speech)*. Komisi Nasional Hak Asasi Manusia, Jakarta, 2015.

[2] G. H. Stanton, "The Rwandan genocide: Why early warning failed," *Journal of African Conflicts and Peace Studies*, vol. 1(2), pp. 6–25, 2009.

[3] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? Predictive features for hate speech detection on twitter," *in Proceedings of the NAACL Student Research Workshop*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 88–93.

[4] I. Alfina, R. Mulia, M. I. Fanany, and Y. Ekananta, "Hate speech detection in the Indonesian language: A dataset and preliminary study," in *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2017, pp. 233–238.

[5] S. B. Shende and L. Deshpande, "A computational framework for detecting offensive language with support vector machine in social communities," in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, July 2017, pp. 1–4.

[6] F. D. Vigna, A. Cimino, F. Dell'Orletta, M. Petrocchi, and M. Esconi, "Hate me, hate me not: Hate speech detection on facebook," in *Proceedings of the First Italian Conference on Cybersecurity (ITASEC17)*, 2017, pp. 86–95.

[7] S. Tulkens, L. Hilte, E. Lodewyckx, B. Verhoeven, and W. Daelemans, "A dictionary-based approach to racism detection in dutch social media," in *First Workshop on Text Analytics for Cybersecurity and Online Safety (TACOS)*, 2016, pp. 11–17.

[8] S. A. Ozel, E. Sarac, S. Akdemir, and H. Aksu, "Detection of cyberbullying on social media messages in Turkish," in 2017 *International Conference on Computer Science and Engineering*, Oct 2017, pp. 366–370.

[9] T. Davidson, D. Warmsley, M. W. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *International AAAI Conference on Web and Social Media (ICWSM)*, 2017, pp. 512–515.

[10] S. Agarwal and A. Sureka, "But I did not mean it!– Intent classification of racist posts on Tumblr," in *2016 European Intelligence and Security Informatics Conference (EISIC)*, Aug 2016, pp. 124–127.

[11] M. Sabou, K. Bontcheva, L. Derczynski, and A. Scharl, "Corpus annotation through crowdsourcing: Towards best practice guidelines," *in Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA), 2014.

[12] C. Goncalves, C. Goncalves, R. Camacho, and E. Oliveira, "The impact of pre-processing on the classification of MEDLINE documents," in *Proceedings of the 10th International Workshop on Pattern Recognition in Information Systems*, 2010, pp. 53–61.

[13] T. Baldwin and Y. Li, "An in-depth analysis of the effect of text normalization in social media." in *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL (HLT-NAACL)*. The Association for Computational Linguistics, 2015, pp. 420–429.

[14] P. C. Gaigole, L. H. Patil, and P. M. Chaudhari, "Preprocessing techniques in text categorization," *IJCA Proceedings on National Conference on Innovative Paradigms in Engineering & Technology 2013*, vol. 3, no. 3, pp. 1–3, December 2013.

[15] I. Kanaris, K. Kanaris, I. Houvardas, and E. Stamatatos, "Words vs. character n-grams for anti-spam filtering," *International Journal on Artificial Intelligence Tools*, vol. 20, no. 10, pp. 1–20, 2006.

[16] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence* - Volume 2, ser. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 1137–1143.

[17] V. Ganganwar, "An overview of classification algorithms for imbalanced datasets," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2(4), pp. 42–47, 2012.

[18] M. Suzuki, N. Yamagishi, Y. Tsai, and S. Hirasawa, "Multilingual text categorization using character n-gram," in *2008 IEEE Conference on Soft Computing in Industrial Applications*, June 2008, pp. 49–54.

[19] B. Plank, "ALL-IN-1: short text classification with one model for all languages," CoRR, 2017.

[20] L. Shi, R. Mihalcea, and M. Tian, "Cross-language text classification by model translation and semi-supervised learning," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 1057–1067.

[21] I. Alfina, S. H. Pratiwi, I. Budi, R. Mulia, and Y. Ekanata, "Detecting hate speech against religion in the Indonesian language," *Journal of Telecommunication, Electronic and Computer Engineering* (JTEC), 2018.

[22] M. O. Ibrohim and I. Budi, "A dataset and preliminaries study for abusive language detection in indonesian social media," *Procedia Computer Science*, vol. 135, pp. 222 – 229, 2018.

[23] A. Bohra, D. Vijay, V. Singh, S. S. Akhtar, and M. Shrivastava, "A dataset of Hindi-English code-mixed social media text for hate speech detection," in *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*. Association for Computational Linguistics, 2018, pp. 36–41.

[24] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, pp. 1–11, 03, 2015.

[25] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *International World Wide Web Conference Committee*, 2017, p. 759760.

[26] E. Sazany and I. Budi, "Deep Learning-Based implementation of hate speech identification on texts in Indonesian: Preliminary study," in *2018 International Conference on Applied Information Technology and Innovation (ICAITI 2018)*, Padang, Indonesia, Sep. 2018.

[27] M. O. Ibrohim, E. Sazany, and I. Budi, "Identify abusive and offensive language in indonesian twitter using deep learning approach," Journal of Physics: Conference Series, 2018.