Toxicity Posts and Hateful Comments Detection in Thai Language Using Supervised Ensemble Classification

Sutthisak Sukhamsri[1*]
[1] Department of Information Technology, Faculty of Science and Agricultural Technology, Rajamangala University of Technology Lanna Tak
41/1 moo 7 Paholayothin Road, Mai Ngam, Mueang, Tak, Thailand, 63000
[*]Corresponding Author: sutthisak@rmutl.ac.th. Phone Number: 08-1499-6516
Received: 7 April 2024, Revised: 25 July 2024, Accepted: 26 July 2024

Abstract
Social media platforms are the community people gather in where they can generally express their free willing opinions to others on any topics they attend. However, on many occasions, the cause of violating arguments or an unpleasant atmosphere in the community is initiated by negative, toxic, and hateful posts or comments. For that reason, monitoring post systems on social media is an essential topic in the natural language processing area, especially in multi-linguistics research. In this study, we proposed a method of improvement for the Thai language's toxic and hateful classification that was trained on the dataset of 2,160 posts from the Thai toxicity Twitter corpus for training and verifying. Therefore, we designated the ensemble approach which includes the combination of XGBoost, multinomial naive Bayes, logistic regression, support vector machine, and random forest for classifiers. In summary, the ensemble classifier improved the previous study in the same dataset with 0.7808 precision, 0.7778 recall, and 0.7721 average accuracies in the weighted F1 scoring with an accuracy of 0.8235 in the F1 binary scoring.
Keywords: Natural Language Processing, Toxicity Posts, Word Vectorization, Thai Language Corpus, Ensemble Model.

1. Introduction

Social media are places where people can express their identity, share their opinions, and idealize. Over 5.18 billion users in late 2023 are engaging with various social media platforms such as Facebook, YouTube, Instagram, TikTok, and X (Twitter's rebranded identity) [1]. Accordingly, numerous users on each social media platform have diverse reasons that persuade them to keep in touch with friends and family, fulfill their spare time, read news, share, and discuss their opinions with the community. So, occasionally some activities on social media platforms may cause a conflict of opposing ideas in a thread of news feeds, influencer sharing, and political issues [2]. Therefore, to identify a negative post or tweet, the linguist attempts to use a taxonomy of word tone in a sentence [3] to classify the toxic or hateful sentence from neutral posts, in which diverse languages also have unique taxonomy and word corpus. Likewise, in the Thai language, many scholars have improved the Thai corpus as NECTEC's ORCHID [4], NECTEC's BEST [5], and Thai user-generated web content (UGWC) [6] corpus.

Observing and identifying hate speech and toxic posts on social media is crucial for ensuring individual well-being and community integrity [7]. These harmful behaviors degrade discourse, create hostile environments, discourage participation, and marginalize vulnerable groups, undermining inclusiveness and democracy. This leads to self-censorship and withdrawal to avoid harassment. The psychological impacts, including stress, anxiety, depression, and suicidal ideation, necessitate effective moderation for mental health protection. Legally, platforms must monitor and remove hate speech to comply with laws and uphold societal norms. Unchecked hate speech can spread misinformation and radicalize individuals, leading to violence and societal destabilization. Economically, toxic environments reduce user engagement, affecting platforms' viability. Addressing these issues fosters safer, more

inclusive, and healthier online communities, promoting respectful dialogue and sustainability.

A significant approach to this problem is to use supervised machine learning techniques to train a model on a labeled dataset of social media posts. The features used in the model can include traditional NLP features like bag-of-words, TF-IDF, or word embeddings, as well as additional features such as sentiment scores, part-of-speech tags, and n-grams. Numerous research papers have proposed specific models to detect hate speech and offensive language on social media. In our study, we proposed a supervised ensemble classification framework for detecting toxic and hateful posts in the Thai language dataset, as shown in section 3, the model results and benchmarks in section 4, and the conclusion of this study in the last section.

2. Related Works

Scholars from Keio University [8] proposed the unigrams and the pattern features as a technique for automatic hate speech detection on Twitter based on the English dataset. Consequently, the pattern features have extracted the unigrams into two categories containing the sentimental word and non-sentimental word for primary unigram features. The dataset contains 7,000 tweets for a training set with three classes for classification prediction such as a hateful class, an offensive class, and a clean class. The various classifiers including the Random Forrest, Support Vector Machine, and J48graft are the candidates for benchmarking, thus the J48graft classifier is outperforming as 0.784 for an F1 accuracy. To sum up, in the same classification model the words feature method has a significance of the model performance. Thus it is the Unigram Features show outperform accuracy followed by the Pattern Features, the Sentiment-based Features, and the Semantic Features.

The study in [9] proposed the word ambiguous manipulation and an automatic sentiment classification for a Thai online document. Accordingly, a combination of deep learning classifiers such as a convolutional neural network (CNN), bi-directional long short-term memory (BLSTM), attention mechanism (ATTN), and bi-directional gates recurrent unit (BGRU) has been selected for performance benchmarking. However, the data preparation for cleaning up a training corpus which includes removing username patterns, emojis, URLs, hashtags, and meaningless characters is a primitive method before document tokenize and word embedding technique to turn a sentence into a vector format before the classification. A collection of 41,073 documents split into 21,490 positive classes and 19,583 negative classes is training and verifying to mentioned deep learning models, found the BGRU+ATTN model performing best result for 91.85% and the others on around 91% F1 scoring accuracy. The deep-learning approach is an appropriate model for classifying a sentiment polarity with a structured vectorizing document even though in the Thai language.

In [10] proposed the comparison of supervised classifiers and deep-learning models for detecting toxic languages in the Thai Twitter dataset. Certainly, this research is separated into two feature extraction techniques, including Bag of Words (BOW) and term frequency-inverse document frequency (TF-IDF). Therefore, the candidate of classifiers including convolutional neural network (CNN), long-short-term memory (LSTM), and pre-trained bidirectional encoder Representations (BERT), were compared with the public Toxicity Thai Twitter corpus. The results show that the Bag of Words (BOW) with the Extra-Tree classifier, has achieved the highest F1-score of 0.72, a classification accuracy rate of 72.27%, and an AUC value of 0.77.

3. Methods

Our study employed three essential methods, incorporating data preprocessing as the preliminary method to input text cleansing and normalizing before vectorizing all documents thus suitable for training and verifying with the classifiers, a brief detail as in sub-sections 3.1, 3.2, and 3.3. The second method is the training and verifying process for all candidate classifiers which all competitors are combined into the ensemble model component and thus is emphasized in sub-section 3.4. The overview of the proposed framework is shown in Figure 1.

3.1 The Dataset

The scholars [11] gathered 3,300 tweets in the Thai language for their study in annotation and classification of toxicity for the Thai Twitter corpus. Therefore, the Thai toxicity tweets corpus is selected from a significant 44 keywords relevant to the Thai toxic words. For example, "จัญไร" (beastly), "สันดาน" (traits), and "ตอแหล" (lie) some of these keywords are common and neutral sentiment semantic words. But, it's likely for offensiveness use depending on the context of the sentences.

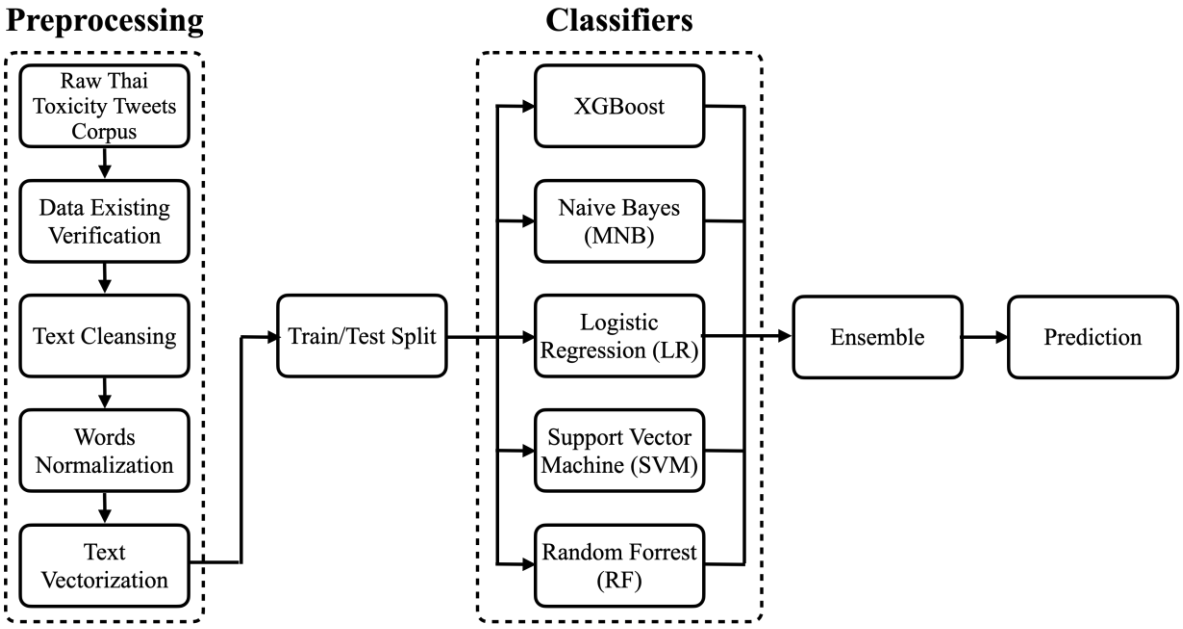

Figure 1. The framework of the ensemble classification method

So, each post has been reviewed and labeled as a toxicity post or neutral post with tripartite labelers for semantic annotation. In brief, for a primitive 3,300 tweets, we found a missing tweet text showing "TWEET_NOT_FOUND" for 506 tweets, along with an empty character in the dataset for 634 tweets. That remains 2,160 tweets for the data cleansing and pre-processing methods, including training with the various machine learning models for identifying a toxicity and neutral post-prediction afterward. The distribution of the trained and tested dataset includes 1,332 posts for toxicity posts with most of the unanimous agreed annotation labeled to class “1”, and 828 posts for neutral tweets as shown in class “0”, thus the tripartite labeled annotation voting proportion as shown in Figure 2. In short, the toxicity posts annotations are labeled with the unanimous agreed referring to 3 votes in tripartite followed by 2 votes are majority agreed, while 1 vote and neglect are labeled as neutral posts

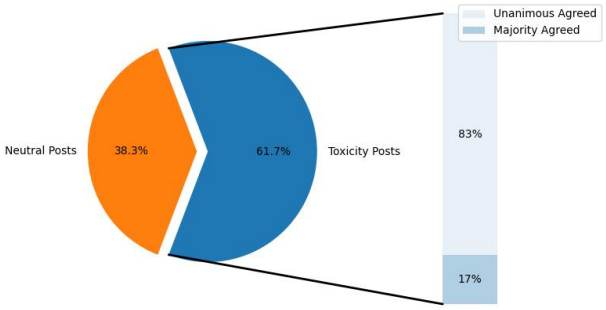

Figure 2. Tripartite Voting Proportion for Toxicity Annotation Posts

3.2 Data Preprocessing

The essential preliminary process for the NLP before the model training is wrangling and cleansing the raw text. Therefore, the text wrangling and the cleansing process is an

arrangement of a primitive text thus retrieved from sources to an appropriate format for an NLP modeling method.

Besides, the advantage of text pre-processing is eliminating an irrelevant context from significant words in the sentence or post. Moreover, the Thai language also contains ambiguous verbal in most Social Media postings, so the word normalizing with the Thai word corpus is the final process for the wrangling and cleansing process. In brief, the pre-processing method involves blank tweet removal, hashtag removal, special characters removal, punctuation & separator removal, URL removal, emoticon & Unicode removal, English words removal, number removal, and word normalization. The details and purpose of the text cleansing process with an instance tweet sample in each step of the processing are shown in Table I. Accordingly, after the preprocessing step, the corpus average words per post would reduce by around 6% from the initial corpus, or around 19.45 words per post after the word normalization process.

3.3 Text Vectorization

Consequently, the last method of the data pre-processing procedure is to vectorize the cleaned document with the term frequency (TF) and inverse document frequency (IDF) method. The TF-IDF is based on the vector space model (VSM) concept to visualize the words in documents to the high dimensional space of its vocabulary, thus represented as a vector. Hence, *TF* is a word *w* frequency count *f* in document *d* is shown in (1), while the *IDF* is the count *N* of document *D* in the corpus where the word term *t* represents in (2). Finally, the normalization of the vector between TF and IDF is as in (3).

$$tf(t,d) = \frac{f(t,d)}{\max\{f(w,d):w \in d\}} \quad (1)$$

$$idf(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|} \quad (2)$$

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D) \quad (3)$$

Table 1. The purpose of text cleansing in the pre-processing methods

Process	Purpose
blank tweet removal	Remove a record by the blank value and the "TWEET_NOT_FOUND" on the "tweet_text" field.
hashtag removal	Remove the character "#" so that Twitter users often post with assigned hashtags relevant to their tweet topic.
punctuation & separator removal	Remove the punctuation characters and excess whitespace created by multiple spaces and tabs.
URL removal	Remove website sharing links thus begin with "http" and "https".
emoticon & Unicode removal	Remove emoticons such as 😉, 👎, ✌, 👆, and 🙏, which include facial expressions and hand gestures, as well as any Unicode emoticons.
English words removal	Remove all English words and single characters, as they are not relevant to the Thai dictionary.
number removal	Remove all numbers from tweet text.
word normalization	Check for spelling errors and correct them based on word similarity and relevance to the Thai dictionary corpus.

3.4 Classifiers

Concerning the classifiers, we had designated aggregate as the baseline model beyond the previous study [10] as mentioned above. Therefore, a brief description of each classifier is shown as follows:

An XGBoost (eXtreme Gradient Boosting) [12] is an algorithm for gradient boosting on decision trees. It is an implementation of the gradient boosting framework that is designed to be highly efficient, flexible, and portable. In text classification, XGBoost can be used to train a model that can predict the class of a piece of text based on its features. The features of a text can be represented using various methods, such as bag-of-words, n-grams, or word embeddings. These features can then be used as input to the XGBoost model, which will learn to predict the class of the text based on its features.

The multinomial Naive Bayes [13] is a variant of the Naive Bayes algorithm that is used for classification problems with discrete features, such as text classification. In a text classification problem, the model takes in a set of documents and their corresponding labels and learns the probability distribution of the words in each class. During the prediction stage, for a new document, the algorithm calculates the likelihood of the document belonging to each class, based on the words it contains, and assigns the class with the highest likelihood as the predicted label.

The Logistic Regression [14] is a supervised learning model that can be used for text classification tasks. The basic idea behind using Logistic Regression for text classification is to convert the text data into a numerical representation, such as bag-of-words or TF-IDF, and then use the numerical representation as input features for the Logistic Regression model. In TF-IDF representation, the text document is represented as a vector of TF-IDF values, which takes into account the frequency of the word in the document and its importance in the corpus.

The Support Vector Machines (SVMs) [15] can be used for text classification tasks such as document categorization and sentiment analysis, in combination with the TF-IDF (term frequency-inverse document frequency) representation of the text data. When training an SVM for text classification, the TF-IDF representation of the tweet text data is used as the input features for the SVM, and the corresponding labels toxic and neutral are used as the output labels. The SVM then learns to separate the different classes of data by finding the hyperplane that separates the different classes in the TF-IDF feature space.

A Random Forest [16] is an ensemble learning method that can be used for text classification tasks. It is an extension of decision trees, and it is composed of multiple decision trees that are trained on different subsets of the data and with different subsets of the features. Following the text classification, a Random Forest algorithm typically works by first converting the text data into a numerical representation, such as TF-IDF or word embeddings.

The Convolutional Neural Networks (CNNs) [17] is a type of deep learning model that can be used for text classification tasks. CNNs are designed to process data that has a grid-like structure, such as images, and they have been adapted to work with text data by treating the text as a grid of words or characters. In text classification, a CNN takes in the text data, which is typically represented as a matrix of word embeddings, where each row corresponds to a word in the text, and each column corresponds to a dimension of the embedding.

An Ensemble Classification, with regards to our proposed supervised soft voting ensemble model as shown in (4) where the ensemble prediction is formed by the arguments of the maxima of weight w and probability p of each classifier which comprises logistic regression, multinomial naive Bayes, the XGBoost, support vector machine, and random forest.

$$y_{ensemble(x)} = \arg\max_i \sum_{j=1}^{n} w_j p_{ij} \quad (4)$$

4. Results

The evaluation process is inaugurated by randomly splitting the dataset to 90:10 for the training and testing dataset. Accordingly, we report the performance of each classifier by adding an amount of each class as the weighted average precision as (5), (6) a weighted average recall as (7), (8) a weighted average F-1 score as (9), (10), and the accuracy as (11) to normalized a minority fluctuation in unbalanced class.

$$\text{Weighted Average Precision} = \frac{|y_0|}{|y|}\cdot P_0 + \frac{|y_1|}{|y|}\cdot P_1 \quad (5)$$

$$P_0 = \frac{T_{P_0}}{T_{P_0}+F_{P_0}} \;;\; P_1 = \frac{T_{P_1}}{T_{P_1}+F_{P_1}} \quad (6)$$

$$\text{Weighted Average Recall} = \frac{|y_0|}{|y|}\cdot R_0 + \frac{|y_1|}{|y|}\cdot R_1 \quad (7)$$

$$R_0 = \frac{T_{P_0}}{T_{P_0}+F_{n_0}} \;;\; R_1 = \frac{T_{P_1}}{T_{P_1}+F_{n_1}} \quad (8)$$

$$\text{Weighted Av. f1 Score} = \frac{|y_0|f1score_0}{|y|} + \frac{|y_1|f1score_1}{|y|} \quad (9)$$

$$\text{f1score}_0 = \frac{2 \cdot P_0 \cdot R_0}{P_0 + R_0} \quad ; \quad \text{f1score}_1 = \frac{2 \cdot P_1 \cdot R_1}{P_1 + R_1} \tag{10}$$

$$\text{accuracy} = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{11}$$

Where y is the set of predicted pairs,
y_n is the set of predicted class n,
P_n is the precision of class n,
R_n is the recall of class n,
T_{P_n} is the true positive of class n,
F_{P_n} is the false positive of class n,
F_{N_n} is the false negative of class n.

In summary, the results are in Table 2. found in a particular candidate classifier that the logistic regression outperformed accuracy in the group at 0.8145 followed by an XGBoost at 0.7895 and the baseline CNN is the third place at 0.7952. Compared with the transformer-based model BERT shown in reference [10] our proposed ensemble model also performed better with around 16% accuracy.

Therefore, the ensemble model is superior and performs in all benchmarks with a 0.7802 in precision accuracy, 0.7778 in recall accuracy, by far the average accuracy of 0.7721, and the overall binary f1-accuracy as 0.8235.

Table 2. The performance of candidate & proposed classifiers

Classifier	**Weighted F1 Accuracy**			***Binary F1***
	Precision	***Recall***	***Average***	***Accuracy***
XGBoost	0.7530	0.7546	0.7518	0.7985
MNB	0.7282	0.7037	0.6757	0.7852
LR	0.7681	0.7639	0.7565	0.8145
SVM	0.7468	0.7407	0.7422	0.7686
RF	0.7303	0.7315	0.7255	0.7852
CNN (Baseline)	0.7652	0.7639	0.7644	0.7952
Ensemble (Proposed)	**0.7802**	**0.7778**	**0.7721**	**0.8235**
BERT [10]	0.6500	0.6800	0.6700	0.6600

5. Conclusion

The ensemble model is an enhancement method for improving the performance by the ability of the candidate classifiers by their prediction and the confidence as a weighted parameter. Particularly, in text mining and document classification applications the basic classifiers have approximate optimal performance in the same range thus the ensemble model is appropriate for overall improvement. As in this research, an ensemble method classifier with TF-IDF featuring methods provides several advantages in text classification tasks. Firstly, they enhance robustness by leveraging diverse representations of text data, improving the model's ability to handle variations in language use and important words in a document structure. Secondly, ensemble classifiers mitigate an overfitting inherent in TF-IDF-based models by aggregating predictions from multiple base classifiers trained on different subsets of the data or feature space. This regularization enhances the model's generalization to unseen documents and reduces the impact of noise in the training data. However, the pre-processing in this research such as text-cleansing is beneficial to eliminate meaningless words and irrelevance punctuation in forming the word vectorization in TF-IDF have a significant to the dimensional reduction for the model training process. In contrast with reference [11] is included emoticons in their training dataset with the belief that they expressed a sentence emotionally. Our experiment in the pre-processing step found the emoticons are useful for sentiment classification, but not quite meaningful in our objective to classify toxicity posts and comments. It is because many sarcastic posts use emoticons as mocking adversative meanings. Hence, we decide to remove all emoticons in the pre-processing task.

Additionally, our proposed methods improve classification accuracy, especially in challenging tasks with imbalanced class distributions or subtle distinctions between document categories. By combining predictions from multiple classifiers, ensemble classifiers achieve higher predictive performance than individual models. Overall, our proposal thus based on an ensemble classifier offers practitioners a powerful approach to building reliable and effective text classification models for toxicity posts and hateful comments detection, which is different from sentiment classification in previous research. The word's meaning is relevant to the tone of the sentences in which the modern language model such as the transformer-based

model or the large language model outperformed for sentimental prediction. But not for toxicity and hate speech detection particularly in written language with connotations like Thai language. So, we found that the sentences vectorized with a thoroughly selected for modeling an ensemble decision are appropriate for classifying a connotative meaning in sarcastic comments based on the degree of word vectorization method in our proposal than the previous study shown as benchmarking in Table 2 results.

Future research and applications for detecting hate speech and toxic posts on social media should enhance algorithmic accuracy and efficiency, leveraging NLP and machine learning model advancements like BERT and GPT. Developing models that understand context, sarcasm, and evolving slang, along with creating multilingual and cross-cultural capabilities by using diverse datasets, is essential. A community-based moderation, where users report harmful content should be explored. Likewise, maintaining trust in automated systems and integrating detection with speech-to-text generation from image and video analysis can enhance. These advancements can be applied across social media platforms, educational research, and fostering safer benefits for more inclusive well-being online communities.

7. References

[1] Kemp S. Digital 2023: Global Overview Report - DataReportal – Global Digital Insights. [cited 2 February 2024]. Available from: https://datareportal.com/reports/digital-2023-global-overview-report.

[2] Yuenyong S, Hnoohom N, Wongpatikaseree K, Ayutthaya TPN. Classification of Tweets Related to Illegal Activities in Thai Language. In: 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP); 2018. p. 1-6.

[3] Mathew B, Kumar N, Ravina, Goyal P, Mukherjee A. Analyzing the hate and counter-speech accounts on Twitter. (Cornell University); 2018.

[4] Sornlertlamvanich V, Takahashi N, Isahara H. Building a Thai part-of-speech tagged corpus (orchid). J Acoust Soc Jpn (E). 1999;20(3):189-198.

[5] Kosawat K, Boriboon M, Chootrakool P, Chotimongkol A, Klaithin S, Kongyoung S, Kriengket K, Phaholphinyo S, Purodakananda S, Thanakulwarapas T, et al. BEST 2009: Thai word segmentation software contest, Natural Language Processing 2009. In: SNLP'09 Eighth International Symposium; 2009. p.83-88.

[6] Lertpiya A, et al. A Preliminary Study on Fundamental Thai NLP Tasks for User-generated Web Content. In: 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP). 2018. p.1-8.

[7] Davidson T, Warmsley D, Macy M, Weber I. Automated hate speech detection and the problem of offensive language. Proceedings of the International AAAI Conference on Web and Social Media. 2017;11(1):512-515.

[8] Watanabe H, Bouazizi M, Ohtsuki T. Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection. IEEE Access. 2018;6:13825-13835.

[9] Piyaphakdeesakun C, Facundes N, Polvichai J. Thai Comments Sentiment Analysis on Social Networks with Deep Learning Approach. In: 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC). 2019. p.1-4.

[10] Thiengburanathum P, Charoenkwan P. A Performance Comparison of Supervised Classifiers and Deep-learning Approaches for Predicting Toxicity in Thai Tweets. In: 2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering. Cha-am, Thailand, 2021. p.238-242.

[11] Sirihattasak S, Komachi M, Ishikawa H. Annotation and classification of toxicity for Thai Twitter. Proceedings of LREC 2018 Workshop and the 2nd Workshop on Text Analytics for Cybersecurity and Online Safety (TA-COS'18). 2018.

[12] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016.

[13] McCallum A, Nigam K, Ungar LH. A comparison of event models for naive Bayes text classification. In: Proceedings of the 15th International Conference on Machine Learning. 1998. p.41-48.
[14] Joachims T. Text categorization with support vector machines: learning with many relevant features. In: European Conference on Machine Learning. Springer, Berlin, Heidelberg. 1998. p.137-142.
[15] Weston J, Chakrabarti S, Weiss Y. Text classification using string kernels. In: Proceedings of the 10th ACM International Conference on Information and Knowledge Management. ACM. November 2001. p.191-198.
[16] Poon H, Domingos P. Random forests for text classification. In: Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence. 2007. p.907-914.
[17] Kim Y. Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. October 2014. p.1746-1751.