Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.035313

Article

Tackling Faceless Killers: Toxic Comment Detection to Maintain a Healthy Internet Environment

Semi Park and Kyungho Lee*

School of Cybersecurity, Korea University, Seoul, 02841, Korea

*Corresponding Author: Kyungho Lee. Email: kevinlee@korea.ac.kr

Received: 16 August 2022; Accepted: 28 September 2022; Published: 09 June 2023

Abstract: According to BBC News, online hate speech increased by 20% during the COVID-19 pandemic. Hate speech from anonymous users can result in psychological harm, including depression and trauma, and can even lead to suicide. Malicious online comments are increasingly becoming a social and cultural problem. It is therefore critical to detect such comments at the national level and detect malicious users at the corporate level. To achieve a healthy and safe Internet environment, studies should focus on institutional and technical topics. The detection of toxic comments can create a safe online environment. In this study, to detect malicious comments, we used approximately 9,400 examples of hate speech from a Korean corpus of entertainment news comments. We developed toxic comment classification models using supervised learning algorithms, including decision trees, random forest, a support vector machine, and K-nearest neighbors. The proposed model uses random forests to classify toxic words, achieving an F1-score of 0.94. We analyzed the trained model using the permutation feature importance, which is an explanatory machine learning method. Our experimental results confirmed that the toxic comment classifier properly classified hate words used in Korea. Using this research methodology, the proposed method can create a healthy Internet environment by detecting malicious comments written in Korean.

Keywords: Toxic comments; toxic text classification; machine learning; healthy internet environment

1 Introduction

Because of the highly contagious coronavirus, the proportion of people working from home has increased 35.4 fold [1]. As their time at home increases, people are spending more time online than offline. The time spent consuming online content has increased by more than 40% compared to the pre-COVID-19 period [2]. However, consequently, toxic comments and cyberbullying have also increased. The anonymity of the Internet has enabled the propagation of hate speech and cyberbullying in the form of offensive, inappropriate, and toxic comments. Cyberbullying causes psychological harm such as depression, trauma, and suicidal tendencies [3–5]. Therefore, online comments are an important topic of research for creating a healthy Internet environment [6–9].

Although hate speech has different legal definitions in different countries, the Cambridge Dictionary defines it as public remarks expressing hate or promoting violence against an individual or group based on race, religion, gender, or sexual orientation. From legal, political, philosophical, and cultural perspectives, hate speech and freedom of expression have some overlapping features. Expressing hate is considered freedom of expression in the United States and some other countries. However, in Germany, the incitement of hatred is punishable under the German Criminal Code. Moreover, in Korea, a person can be punished under criminal law if someone defames, abuses, or insults another individual. Owing to its unique cultural characteristics, all countries have different views on hate speech. Waldron [10] argued that the expression of hate should be regulated because it has severe consequences on the lives, dignity, and reputations of members of minority groups.

Owing to the recent Korean wave, including the drama series Squid Games and the K-pop boy band BTS, the monetary value of exported Korean content (K-content) exceeded more than 10 billion dollars [11]. This Korean wave resulted in the dissemination of quickly translated Korean content to the world, but malicious comments and fake news were also exported. Malicious comments are increasingly becoming a social and cultural concern; therefore, the detection of offensive comments in advance at the national level and malicious users at the corporate level is critical. However, Parekh et al. [12] found that studies on other languages are lacking compared to studies on English. Current technology used to detect malicious comments is focused on the English language [13]. To detect harmful information on the Internet in advance at the national and corporate levels, it is critical to study malicious comments or hate expressions among Korean words. Machine or deep learning methods have been used to detect malicious content in many social media posts or news comments.

In a previous study, a novel approach to automatically classifying toxic comments and preventing them from being posted was proposed [14–17]. Machine learning methods have been used in several studies on detecting harmful comments containing profanity, hate speech, toxic speech, and extremism [18–20]. Researchers have been using conventional machine learning algorithms, such as decision tree (DT), logistic regression, and support vector machine (SVM) models. Furthermore, neural network algorithms, such as convolutional neural networks (CNN) and long short-term memory (LSTM) have been developed. Through the Kaggle challenge, Google is developing tools for detecting toxic comments and ensuring healthy online conversations, including the development of a multi-label classifier used to detect harmful comments, including threats, obscenities, and profanity [21]. However, no tool can reliably discriminate between a low false-positive rate and high accuracy [18].

Toxic comments are intended to maliciously demean people. This study contributes to research on natural language processing (NLP) in Korean for the classification of toxic comments. In this study, we used a supervised learning-based classifier to detect hateful words in Korean. In addition, using a Korean online news comment corpus, we developed a classifier to determine whether a comment is toxic. We used a random forest algorithm and achieved an F1-score of 0.94. We also used the feature importance to attain model interpretability, thereby contributing to the development of a healthy Internet culture.

The remainder of this paper is organized as follows. In Section 2, related works on the detection of toxic comments are reviewed. The proposed methodology and data preparation process are described in Section 3. Section 4 presents some experimental results. Finally, Section 5 provides some concluding remarks and areas of future research.

2 Related Work

Machine and deep learning methods have been used to detect malicious content in many social media posts or news comments. Using a perspective approach, Google's Jigsaw conducted a study on the automatic detection of harmful language on social media platforms using machine learning [21]. A study was conducted to neutralize the toxicity detection system through hostile cases in which the perspective API and the project output are deceived. The perspective score can be lowered by modifying the toxic phrase through the use of space. Duplicating characters and periods can also help lower the toxicity score. The Wikipedia dataset provided by Kaggle during the Toxic Comment Classification Challenge contains 159,571 records for classifying malicious comments. This dataset has a class imbalance, and most of it was written in English because it consists of data collected from English Wikipedia. High accuracy was achieved in classifying the toxicity in this dataset through the use of linear regression (LR), a CNN, an LSTM, a gated recurrent unit, and CNN + LSTM [18–20].

Yin et al. [14] created a dataset of 24,783 tweets using the Twitter API. They classified these tweets into offensive, clean, and hated classes. Most of the dataset consisted of offensive Twitter messages (tweets). Various detection tools have been developed to automatically identify harmful messages [15,16]. Nobata et al. [16] proposed a methodology for detecting hate speech that is subtler than profanity in various settings by improving the level of knowledge using a deep learning method. Chen et al. [17] proposed a lexical syntactic feature architecture for detecting offensive language on social media and achieved an accuracy of 98.24% in detecting sentence attacks and 77.9% accuracy in detecting user attacks.

Although many toxicity detection studies have been conducted in English, few studies have focused on the Korean language; therefore, harmful language detection in related Korean datasets is required. A total of 9,400 online news comments in the Korean Hate Speech Dataset were collected and manually labeled [22]. Park et al. [23] developed a dataset of 100,000 people by collecting unlabeled data from public datasets. online communities, and news portal comments. The model was trained using a Bi-LSTM neural network model and generative pre-training, and by efficiently learning an encoder (ELECTRA) that accurately classifies the token replacements, and achieved excellent performance with an F1-score of 0.963 and a mean squared error of 0.029. Park et al. [24] proposed the Korean Text Offensiveness Analysis System (KOAS) to measure profanity and implicit aggression for the analysis of insults in Korean. A total of 46,853 Korean sentences from three domains were collected and classified as positive, neutral, or hostile. The KOAS revealed proficiency detection accuracy of more than 90% and an emotion analysis accuracy of more than 80%. We propose a machine learning based classifier that understands the characteristics of Korean culture and analyzes Korean characteristics for detecting malicious comments.

3 Methodology

As of 2020, Korea has an Internet usage rate of 96.5%, and the majority of people use the Internet [25]. Koreans have created diverse online communities to share news and small stories. Although an online community is a place of connection that provides useful information and allows people to comfort each other, it also becomes a platform for disseminating slander, insults, and false information online under the guise of freedom of expression. In Korea, defamation and insults on the Internet are stipulated in the Criminal Act, allowing a complaint to be filed, or an accusation to be leveled against an offending party. Platform operators typically share the personal information of users who have posted malicious comments on the community to assist with an investigation into such cases. Thus, personal information is provided to investigative agencies, and the user churn rate for the platforms increases. As personal information is provided to investigative agencies, platform users will leave the community. To prevent this, operators have introduced a function for automatically expressing swear words in an alternative form when users post a comment, or a way to detect and delete malicious comments. In addition, after some celebrities committed suicide following the posting of malicious comments, Naver and Daum, the largest Korean online platforms, prohibited users from writing comments on entertainment news articles.

This study was conducted to detect toxic comments in Korean, and the detailed methodology is shown in Fig. 1. The Korean hate speech dataset (KHSD) was created by collecting Korean news comments. This dataset is the first reliable, manually annotated corpus dataset collected in the Korean language. In this study, only nouns were extracted using the kind Korean morpheme analyzer (KKMA) preprocessing library, and an exploratory data analysis was conducted using a holdout validation. Subsequently, vectorization was conducted using the term frequency-inverse document frequency (TF-IDF) for NLP. A data analysis model was created for evaluating classification algorithms, including K-nearest neighbor (K-NN) and random forest (RF).

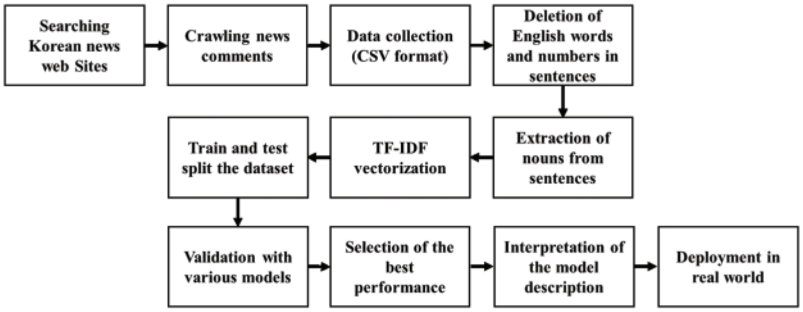

Searching Korean news web Sites
Crawling news comments
Data collection (CSV format)
Deletion of English words and numbers in sentences
Extraction of nouns from sentences
TF-IDF vectorization
Train and test split the dataset
Validation with various models
Selection of the best performance
Interpretation of the model description
Deployment in real world

Figure 1: Flowchart of research methodology

3.1 Dataset

The Korean Hate Speech Dataset is a Korean online news comment corpus that detects prejudice, hate speech, and insults. The criteria for judging malicious comments differ considerably for each person and extracting their harmful elements is difficult. This dataset was created to introduce automation when many insults related to malicious comments were introduced [22].

In the dataset, 32 annotators participated in the pilot study and final tagging processes to investigate whether each comment expressed social prejudice and hatred. Social prejudice was categorized into opinions on gender, other opinions, and none, and hate prejudice was categorized into hate, aggression, or none. This dataset was manually annotated and consisted of 7,896 training data, 471 validation data, and 974 test data. We checked three conditions to classify the data as hateful. First, we checked whether a misrepresentation or insults are content that expresses hostility toward a particular group. Second, we checked whether the content is an expression of hostility toward an individual or a statement that seriously undermines one's social status. Third, we checked whether hate speech or insults are present. We also classified a comment as offensive if it was a sarcastic, cynical, crosstalk, or inhumane remark that offended the target of the comment or any third parties who viewed it.

To create a toxic comment classification model, this study defined all offensive expressions as toxic.

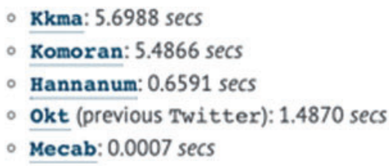
3.2 Preprocessing

3.2.1 Korean Language Preprocessing (KoNLPy)

In this study, a different method was used to preprocess English words because the KHSD, which is a Korean corpus, was applied. Because different morphemes and suffixes can be connected to the same word in Korean, if word-based counting used in English or other languages is attempted, it is not treated as the same word. Although the meaning is the same, it is vectorized into a different word, which makes natural language processing in Korean ineffective. For this reason, it must be preprocessed differently from English. Thus, strings containing unnecessary information were removed using regular expressions that are almost similar. In the case of Hangul, the range of consonants is from "ㄱ" to "ㅎ," and the range of vowels is from "ㅏ" to "ㅣ." To combine these characters and generally leave only Hangul characters, the range of Hangeul is from "가" to "힣." In this study, the rest of the range, other than Hangul characters, was replaced with spaces. Similar to the abbreviation "lol" in English, Korean also contains shortened phrases consisting of consonants only, such as "ㅋ ㅋ" and "ㅎ ㅎ." However, to the best of our knowledge, no practical benefit exists, and even if analyzed, these consonants exist in most comments and are all replaced with blanks because no significant use exists in detecting malicious expressions. Thereafter, only nouns were extracted using the KoNLPy tagging class through a Korean preprocessing library called KKMA [26].

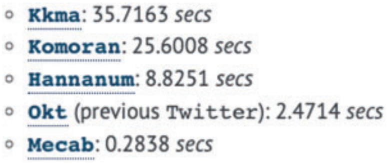
The KKMA used in this study works well regardless of spacing errors. KKMA uses dynamic programming to handle situations in which one or more possible morphemes appear from a single syllable. Considering the number of cases in which the length of a word is increased from one syllable, possible combinations of morphemes at a specific length are created and stored in memory. Then, by increasing the length sequentially again, it is possible to consider all possible combinations of morphemes by generating a new result based on the previously stored results. A dictionary-based morpheme combination is used to determine whether a morpheme combination created using dynamic programming is suitable. Part-of-speech combination, phonemic contention, and form combination are example conditions. A probabilistic model is used to select the candidate group that meets such conditions. Through the above method, KKMA generates several candidate morpheme combinations through dynamic programming and an adjacency condition check, and finally separates the stems and endings through a probabilistic model.

KoNLPy is a Python package for representative Korean NLP. The Korean language is the 13th most spoken language in the world, and various tools, such as KoNLPy, have been developed to extract valuable characteristics from texts based on their complexity and subtlety. KoNLPy is open-source software that can adopt five tagging methods internally. Tagging denotes analyzing of morphemes in Korean, which refers to grasping the structure of various linguistic attributes, such as morphemes, roots, prefixes, suffixes, and parts of speech. As shown in Figs. 2 and 3, based on the documentation of the KoNLPy project, KKMA exhibits slower loading and execution time compared with other tagging classes.



- Kkma: 5.6988 *secs*
- Komoran: 5.4866 *secs*
- Hannanum: 0.6591 *secs*
- Okt (previous Twitter): 1.4870 *secs*
- Mecab: 0.0007 *secs*

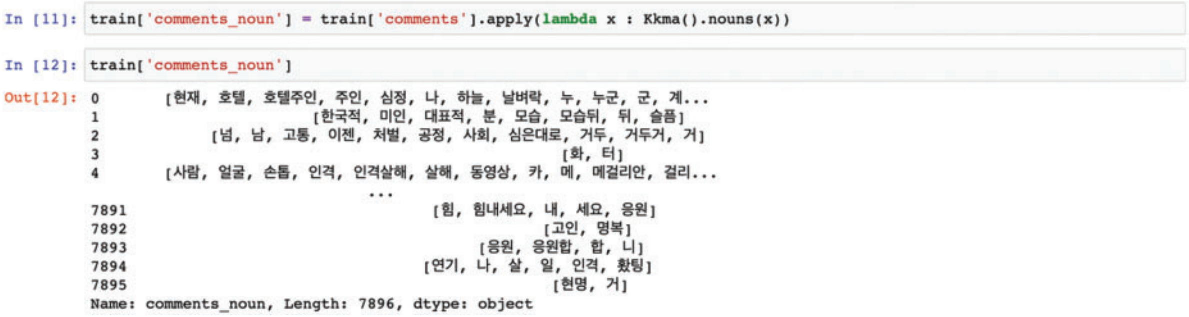
Figure 2: Loading time of KoNLPy tagging class



- Kkma: 35.7163 *secs*
- Komoran: 25.6008 *secs*
- Hannanum: 8.8251 *secs*
- Okt (previous Twitter): 2.4714 *secs*
- Mecab: 0.2838 *secs*

Figure 3: Execution time of KoNLPy tagging class

If the dataset size is large, another tagging class should be considered because of the time problem of the KKMA. Because the dataset used in our experiment is not large, we performed the experiment using the KKMA tagging class. Compared with Hannanum, Komoran, Mecab, and Okt, the tagged results of this study were used to obtain the performance data we wanted in KKMA; therefore, we conducted an experiment using the data shown in Fig. 4.



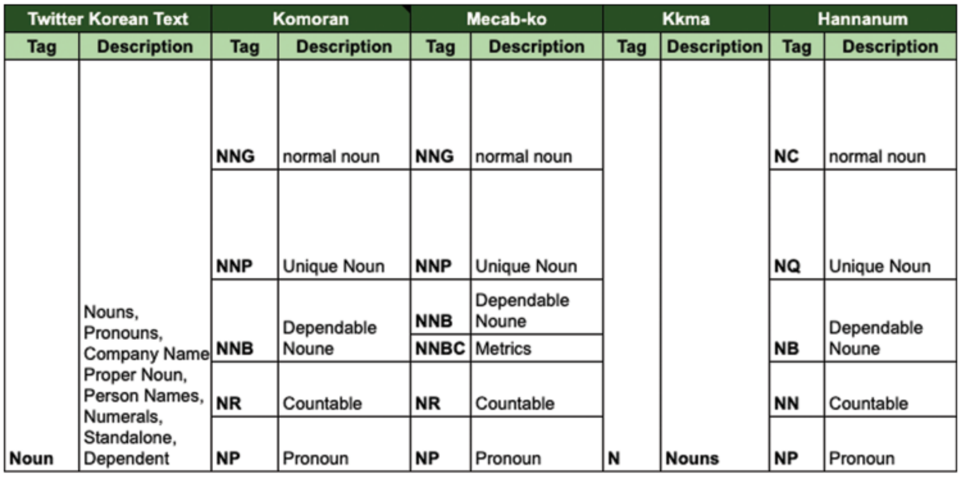
```
In [11]: train['comments_noun'] = train['comments'].apply(lambda x : Kkma().nouns(x))

In [12]: train['comments_noun']

Out[12]: 0       [현재, 호텔, 호텔주인, 주인, 심정, 나, 하늘, 날벼락, 누, 누군, 군, 계...
         1                         [한국적, 미인, 대표적, 분, 모습, 모습뒤, 뒤, 슬픔]
         2             [넘, 남, 고통, 이젠, 처벌, 공정, 사회, 심은대로, 거두, 거두거, 거]
         3                                                             [화, 터]
         4       [사람, 얼굴, 손톱, 인격, 인격살해, 살해, 동영상, 카, 메, 메걸리안, 걸리...
                                      ...
         7891                                  [힘, 힘내세요, 내, 세요, 응원]
         7892                                              [고인, 명복]
         7893                                     [응원, 응원합, 합, 니]
         7894                                 [연기, 나, 살, 일, 인격, 홨팅]
         7895                                              [현명, 거]
         Name: comments_noun, Length: 7896, dtype: object
```

Figure 4: Results of KKMA tagging class

The noun used in KKMA matches the tagging result of another tagging class as displayed in Fig. 5.



Twitter Korean Text		Komoran		Mecab-ko		Kkma		Hannanum	
Tag	Description	Tag	Description	Tag	Description	Tag	Description	Tag	Description
Noun	Nouns, Pronouns, Company Name Proper Noun, Person Names, Numerals, Standalone, Dependent	NNG	normal noun	NNG	normal noun	N	Nouns	NC	normal noun
		NNP	Unique Noun	NNP	Unique Noun			NQ	Unique Noun
		NNB	Dependable Noune	NNB	Dependable Noune			NB	Dependable Noune
				NNBC	Metrics				
		NR	Countable	NR	Countable			NN	Countable
		NP	Pronoun	NP	Pronoun			NP	Pronoun

Figure 5: None tagging rule

3.2.2 Sentence Vectorization (TF-IDF)

TF-IDF is a method of weighing the importance of each word using word and inverse document frequencies [27]. TF-IDF can not only statistically express how often a specific word appears in a document but also compare the weights of the words in the document and express similarity between documents using the cosine similarity method [28,29].

$$tf\,(\mathrm{d},\mathrm{t}) \tag{1}$$

$$df\,(t) \tag{2}$$

$$idf\,(d,t) = \log\left(\frac{n}{1+df\,(t)}\right) \tag{3}$$

$$tf\text{–}idf = tf\,(\mathrm{d},\mathrm{t}) * idf\,(d,t) = tf\,(\mathrm{d},\mathrm{t}) * \log\left(\frac{n}{1+df\,(t)}\right) \tag{4}$$

Eq. (1) represents the number of occurrences of a specific word t in a specific document *d*; thus, a value representing the frequency of occurrence of each word in each document can be expressed. Eq. (2) represents the number of documents in which a specific word t appears. As the number of documents appearing for a word containing a specific topic is tracked, this number is a critical factor in determining the rarity and importance of a specific word *t*. The *idf* value is inversely proportional to Eq. (3); therefore, this device prevents the weights from overflowing in the model for technical terms, slang, and misspelled words. When the value of $df\,(t)$ becomes zero, the value of $tf\,(d,t)$ in Eq. (4) can also be ignored. In Eq. (4), the final value is determined by multiplying the value of $tf\,(d,t)$ by the value of $idf\,(d,t)$ by reflecting the aforementioned series of processes [30].

Because of the usefulness as discussed, TF-IDF was applied as a baseline model instead of a one-hot encoding method that is discontinuously expressed when conducting NLP classification tasks.

3.3 Data Analysis Model

TF-IDF was used for vectorization, and a classification algorithm was used for classifying online news comments. Representative classification algorithms include decision tree, random forest, support vector machine, and K-NN algorithms, as shown in Fig. 6.

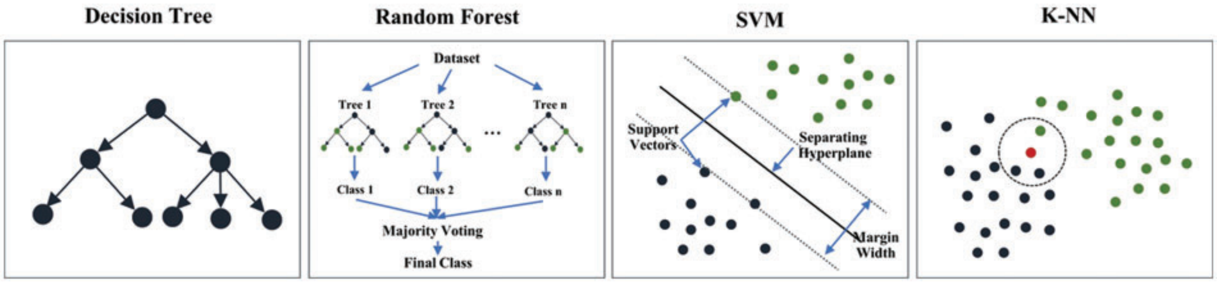


Figure 6: Decision tree, random forest, SVM, and K-NN

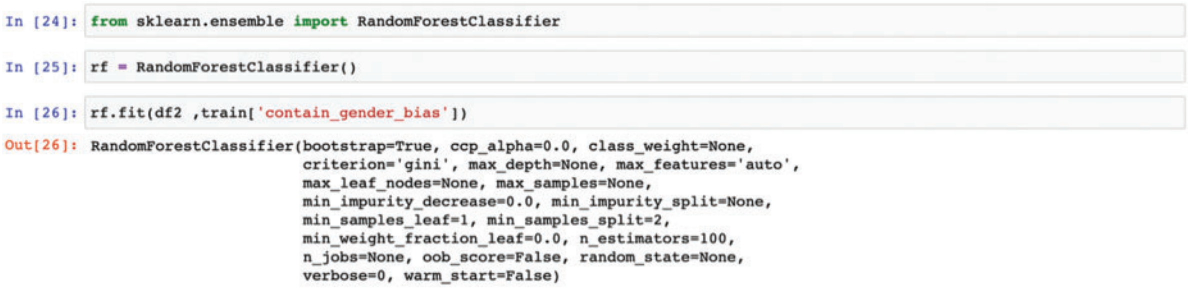
A DT is a tree that sorts and categorizes objects based on the shape values. Each node represents the shape of the object to be categorized, and each branch represents a possible value for the node. It starts with the root node, and their feature values are arranged into an instance. A DT is commonly used to categorize data in a variety of computational domains. Tree paths or rules are mutually exclusive and complete, which is an interesting and important property of decision trees and rule sets.

RF is a classification algorithm composed of several decision trees and is used to determine the most appropriate classification in a subset. Many trees are created and adjusted to solve the overfitting problem of the decision trees; therefore, they do not considerably affect the prediction. Because each of its nodes reflects the shape of an instance to be classed and each branch indicates a value that the node can assume, a DT sorts and categorizes instances based on the shape values. According to the feature values, instances are sorted, starting with the root node. DTs are primarily used in various computational fields to classify data. DT learning algorithms are widely accepted because they can be applied to numerous problems. The fact that tree routes or rules are mutually exclusive and complete is an exciting and essential property of decision trees and rule sets [31].

SVMs are supervised learning algorithms that can be used for classification, regression, and outlier detection. Because an SVM has a high-dimensional space, a subset of the training points is employed in the decision function for specifying the memory efficiency and final kernel functions for the decision function. Although a standard kernel is provided, a custom kernel can be specified. Most real-world problems involve indivisible data, i.e., data in which no hyperplane separates positive from negative instances in the training dataset. This separability problem can be resolved by mapping the data to a high-dimensional space and defining a split hyperplane. The modified feature space, in contrast to the input space filled in by the training instance, has a large number of dimensions [32].

A K-NN is a final example. When there is little or no prior knowledge of the data distribution, K-NN classification is one of the most extensively used methods for classifying objects. When reliable parameter estimates of the probability density are unknown or difficult to determine, a K-NN is a viable choice for achieving a discriminant analysis. A K-NN is a supervised learning technique that classifies the results of a new instance query using the majority of the k-parameter neighbor categories. The main task of the algorithm is classifying new entities using attributes and training data. In this case, a majority vote among k items is used for the classification.

We compared the performances of the DT, RF, SVM, and K-NN algorithms. Scikit-learn was used to implement the four models, and the default values provided by scikit-learn were applied as the hyperparameters, as shown in Fig. 7.



```
In [24]: from sklearn.ensemble import RandomForestClassifier

In [25]: rf = RandomForestClassifier()

In [26]: rf.fit(df2 ,train['contain_gender_bias'])

Out[26]: RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                                criterion='gini', max_depth=None, max_features='auto',
                                max_leaf_nodes=None, max_samples=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, n_estimators=100,
                                n_jobs=None, oob_score=False, random_state=None,
                                verbose=0, warm_start=False)
```

Figure 7: Code of RF classifier

3.4 Exploratory Data Analysis

The data were constructed using the holdout validation. Although the test set should be used to evaluate the model performance, the test set of the dataset used in our experiments cannot be applied because it is used in Kaggle competitions [22]. Therefore, the validation set was used as the test set for validation and the training set for model training. The dataset consisted of 6,664 normal (false) news comments and 1,232 malicious (true) comments, as shown in Fig. 8.

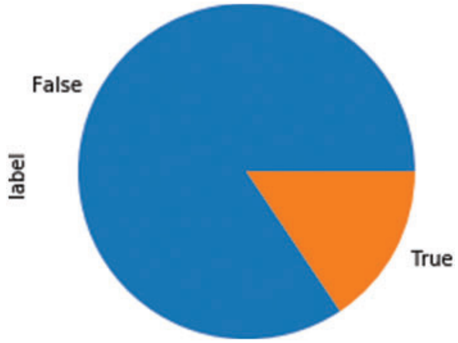


Figure 8: Label distribution of dataset

Fig. 9 shows the distribution of the dataset after the first step of preprocessing, i.e., after string removal using the regular expressions. Most of the comments were within 40 characters.

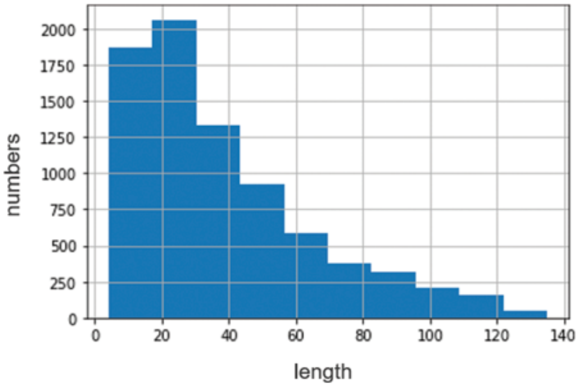


Figure 9: Length distribution of the dataset

After extracting the noun part, which is the second step of the preprocessing, the distribution by the label was evaluated, as displayed in the graph in Fig. 10. For the words in these comments, the difference in the distribution between malicious and normal comments could not be confirmed. The experiment revealed that maliciousness should be classified according to the actual content of the comment rather than the statistical characteristics, including sentence length and number of words.

This technique was applied to the test set in the same manner as the method applied in the data preprocessing of the training set. A TF-IDF vectorizer was used to vectorize the dataset. The vectorized results are displayed in Fig. 11.

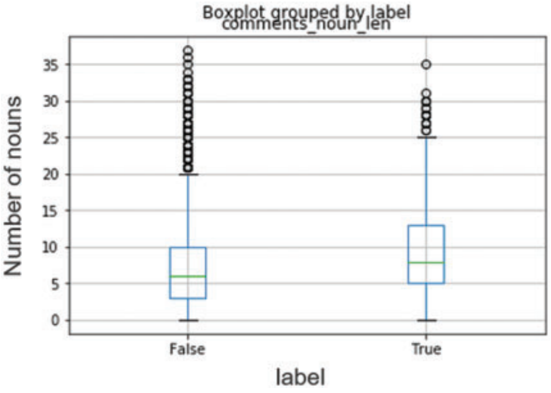
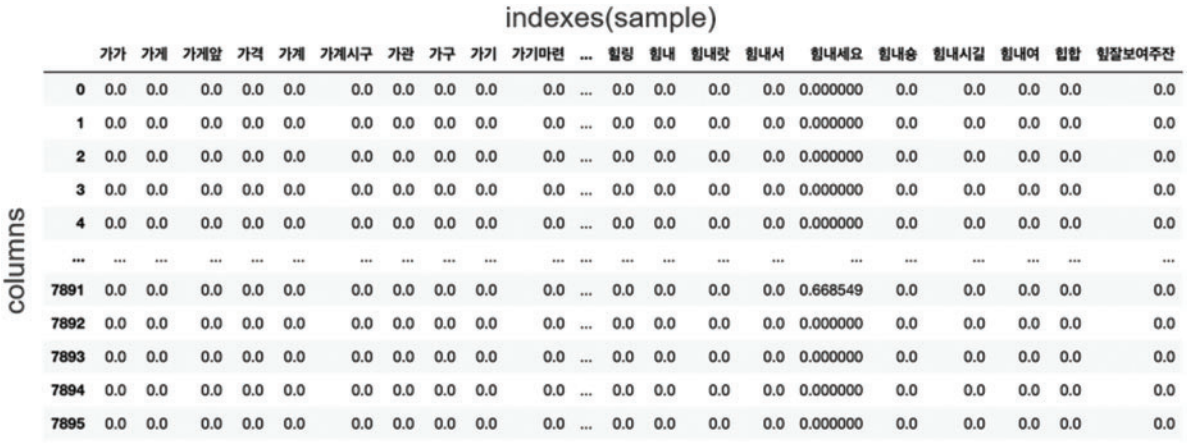


Figure 10: Boxplot grouped by label comments *vs.* number of nouns


indexes(sample)

columns

	가가	가게	가게앞	가격	가계	가계시구	가관	가구	가기	가기마련	...	힐링	힘내	힘내랏	힘내서	힘내세요	힘내숑	힘내시길	힘내여	힙합	힘잘보여주잔
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7891	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.668549	0.0	0.0	0.0	0.0	0.0
7892	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
7893	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
7894	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
7895	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0

Figure 11: Vectorized comments using TF-IDF

4 Results

4.1 Model Validation

The trained model predicted the test data and evaluated its accuracy. The precision, recall, accuracy, and F1-score were used as representative metrics. We used these metrics shown in Fig. 12 for model validation. The precision is the ratio of true negatives *vs.* true positives, whereas the recall is the ratio of false positives *vs.* true positives. The accuracy is intuitively related to the model performance. The F1-score is a harmonic average value that considers both precision and recall.

These values can be expressed as follows:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$F1\text{–}Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8}$$

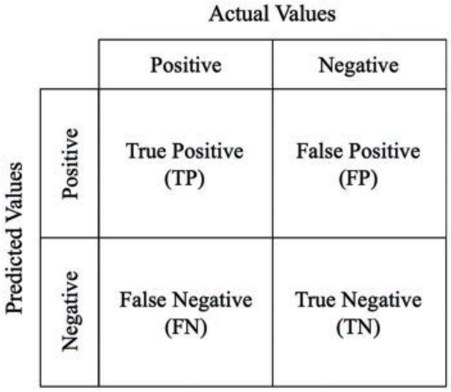


Figure 12: Confusion matrix

We used classification_report of scikit-learn to obtain the results of the four metrics in Table 1. As a results of training the DF, RM, SVM, and K-NN models for predicting the test data, the RF model was shown to be excellent based on all scores.

Table 1: Evaluation results of different classifiers

Classifier	Precision	Recall	Accuracy	F1-score
Decision tree	0.90	0.90	0.90	0.90
Random forest	0.94	0.94	0.94	0.94
SVM	0.92	0.92	0.92	0.91
K-NN	0.78	0.86	0.86	0.79

4.2 Model Description

Even if the trained model achieves an excellent performance, evaluating whether an overfitting problem exists is crucial. Therefore, the concepts of explanatory AI (XAI) and permutation feature importance, which can be used to identify which features are important, have emerged [33]. The degree of error of the model is measured by removing each feature and replacing the removed feature with random noise to eliminate the relationship with the resulting value. We can confirm that if there is a difference in error, the feature is important when the model makes a prediction. The importance of the permutation function is calculated as follows:

The inputs are the fitted predictive model m and tabular dataset D. The training dataset is input into D in our experiment. The reference score of model m on data D is computed as s. For example, the score is calculated using the accuracy for a classifier or R^2 for a regressor. For each feature j (column in D) and for each repetition K of $(1, \ldots, K)$, a corrupted version of data $\widetilde{D_{k,j}}$ is generated by randomly shuffling the column j of dataset D. The score $s_{k,j}$ is computed for model m on the corrupted data, $\widetilde{D_{k,j}}$. Eq. (9) is the formula for computing importance i_j for feature f_i [34]:

$$i_j = s - \frac{1}{K}\sum\nolimits_{k=1}^{K} s_{k,j} \tag{9}$$

This study confirmed that the RF achieved the best performance among the other approaches. The RF provides interpretability through its feature importance. The results of XAI using permutation feature importance are displayed in Fig. 13.

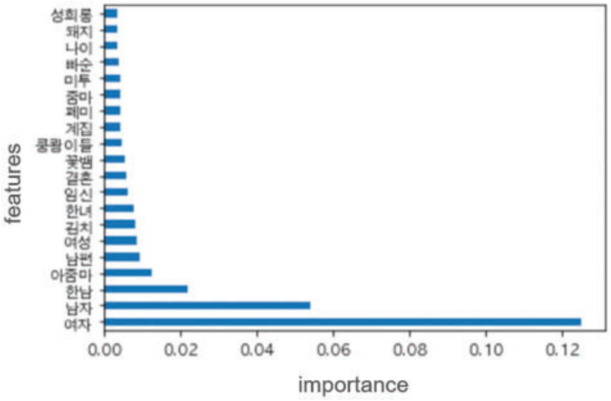


Figure 13: Feature importance results

Here, "돼지" (pigheaded), "빠순" (fangirl), "한남" (Korean men), and "한녀" (Korean girl) displayed in Fig. 13 are representative of hate speech words used in Korea. As a result of feature importance, the learning was conducted well because the corresponding hate expressions appeared as actual weights in the model.

5 Conclusion

Online trolls claim the right to freedom of expression, make light of hate speech, and post controversial comments on the Internet. Troll comments propagate form individuals to society, and campaigns have been conducted worldwide. The No Hate Comments Day campaign was held with the participation of teenagers from ten countries, including the United States and Australia. The Kind Comments campaign was created to protect youth from cyberbullying and prevent suicides by celebrities resulting from mean comments. Although we did not directly participate in this campaign, we had a small hand in creating a healthy Internet environment. Thus, we studied the detection of offensive comments using machine learning.

In this study, a harmful comment classification model was generated using the KHSD, which is a Korean news comment corpus. Nouns were extracted using KoNLPy that Korean preprocessing library and were subsequently vectorized using TF-IDF. We proposed a toxicity detection model using DT, RF, SVM, and K-NN for classification. As a result of the experiment, the F1-score of the RF algorithm was 0.94, which was the highest performance achieved.

This study obtained the decision criteria in which toxic words were classified by the proposed model through XAI. The feature importance of the trained model was analyzed to prove that hate expressions commonly used in Korea have a high weight in our experiment.

The limitations of our study were the use of only two algorithms and one dataset because reliable Korean comment datasets have not been sufficiently developed. In the future, we plan to apply various classification algorithms to various datasets. Moreover, we expect that researchers will use deep learning to detect new Korean hate words.

Funding Statement: This study was supported by a grant-in-aid of HANWHASYSTEMS.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Shiftee, “Shiftee reveals WFH big data analysis of 2020 and 2021,” 2022. [Online]. Available: https://shiftee.io/en/blog/article/shifteeWorkFromHomeBigDataNews

[2] Korea Research, “[Plan] malicious comments, whether regulation and blocking are the best–A study on the perception of comments,” 2021. [Online]. Available: https://hrcopinion.co.kr/archives/17398

[3] S. Hinduja and J. W. Patchin, “Bullying, cyberbullying, and suicide,” *Archives of Suicide Research*, vol. 14, no. 3, pp. 206–221, 2010.

[4] S. Alhabash, A. R. McAlister, C. Lou and A. Hagerstrom, “From clicks to behaviors: The mediating effect of intentions to like, share, and comment on the relationship between message evaluations and offline behavioral intentions,” *Journal of Interactive Advertising*, vol. 15, no. 2, pp. 82–96, 2015.

[5] M. Hsueh, K. Yogeeswaran and S. Malinen, “Leave your comment below: Can biased online comments influence our own prejudicial attitudes and behaviors?,” *Human Communication Research*, vol. 41, no. 4, pp. 557–576, 2015.

[6] M. Koutamanis, H. G. Vossen and P. M. Valkenburg, “Adolescents’ comments in social media: Why do adolescents receive negative feedback and who is most at risk?,” *Computers in Human Behavior*, vol. 53, pp. 486–494, 2015.

[7] M. J. Lee and J. W. Chun, “Reading others’ comments and public opinion poll results on social media: Social judgment and spiral of empowerment,” *Computers in Human Behavior*, vol. 65, pp. 479–487, 2016.

[8] H. Rim and D. Song, “How negative becomes less negative: Understanding the effects of comment valence and response sidedness in social media,” *Journal of Communication*, vol. 66, no. 3, pp. 475–495, 2016.

[9] L. Rösner, S. Winter and N. C. Krämer, “Dangerous minds? Effects of uncivil online comments on aggressive cognitions, emotions, and behavior,” *Computers in Human Behavior*, vol. 58, pp. 461–470, 2016.

[10] J. Waldron, “Approaching hate speech,” in *The Harm in Hate Speech*, Cambridge, MA and London, England: Harvard University Press, pp. 1–17, 2012.

[11] The Financial News, “K-content exports exceed 14 trillion won in the ‘Korean Wave’,” 2022. [Online]. Available: https://www.fnnews.com/news/202201240917248024

[12] P. Parekh and H. Patel, “Toxic comment tools: A case study,” *International Journal of Advanced Research Computer Science*, vol. 8, no. 5, pp. 964–967, 2017.

[13] J. A. Leite, D. F. Silva, K. Bontcheva and C. Scarton, “Toxic language detection in social media for Brazilian Portuguese: New dataset and multilingual analysis,” arXiv preprint arXiv:2010.04543, 2020.

[14] D. Yin, Z. Xue, L. Hong, B. D. Davison, A. Kontostathis *et al.,* “Detection of harassment on web 2.0,” in *Proc. Content Analysis in the WEB*, Madrid, Spain, vol. 2, pp. 1–7, 2009.

[15] C. van Hee, G. Jacobs, C. Emmery, B. Desmet, E. Lefever *et al.,* “Automatic detection of cyberbullying in social media text,” *PLoS One*, vol. 13, no. 10, pp. e0203794, 2018.

[16] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad and Y. Chang, “Abusive language detection in online user content,” in *Proc. 25th Int. Conf. World Wide Web*, Montreal, Canada, pp. 145–153, 2016.

[17] Y. Chen, Y. Zhou, S. Zhu and H. Xu, “Detecting offensive language in social media to protect adolescent online safety,” in *2012 Int. Conf. Privacy, Security, Risk and Trust and 2012 Int. Conf. on Social Computing*, Amsterdam, Netherlands, pp. 71–80, 2012.

[18] S. Zaheri, J. Leath and D. Stroud, “Toxic comment classification,” *SMU Data Science Review*, vol. 3, no. 1, pp. 13, 2020.

[19] M. A. Saif, A. N. Medvedev, M. A. Medvedev and T. Atanasova, "Classification of online toxic comments using the logistic regression and neural networks models," in *AIP Conf. Proc.*, Sozopol, Bulgaria, vol. 2048, no. 1, pp. 060011, 2018.

[20] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis and V. P. Plagianakos, "Convolutional neural networks for toxic comment classification," in *Proc. 10th Hellenic Conf. Artificial Intelligence*, Patras, Greece, pp. 1–6, 2018.

[21] H. Hosseini, S. Kannan, B. Zhang and R. Poovendran, "Deceiving google's perspective API built for detecting toxic comments," arXiv preprint arXiv:1702.08138, 2017.

[22] J. Moon, W. I. Cho and J. Lee, "BEEP! Korean corpus of online news comments for toxic speech detection," arXiv preprint arXiv:2005.12503, 2020.

[23] J. W. Park, Y. Y. Na and K. Park, "A new dataset for Korean toxic comment detection," in *Proc. Korea Information Processing Society Conf.*, Yeosu, Korea, pp. 606–609, 2021.

[24] S. H. Park, K. M. Kim, S. Cho, J. J. Park, H. Park *et al.,* "KOAS: Korean text offensiveness analysis system," in *Proc. 2021 Conf. Empirical Methods in Natural Language Processing: System Demonstrations*, Punta Cana, Dominican Republic, pp. 72–78, 2021.

[25] International Telecommunication Union, "ITU releases 2021 global and regional ICT estimates," 2021. [Online]. Available: https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx

[26] E. L. Park and S. Cho, "KoNLPy: Korean natural language processing in Python," in *Annu. Conf. Human and Language Technology*, Gangwon-do, Korea, pp. 133–136, 2014.

[27] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[28] H. C. Wu, R. W. P. Luk, K. F. Wong and K. L. Kwok, "Interpreting TF-IDF term weights as making relevance decisions," *ACM Transactions on Information Systems (TOIS)*, vol. 26, no. 3, pp. 1–37, 2008.

[29] S. Tata and J. M. Patel, "Estimating the selectivity of TF-IDF based cosine similarity predicates," *ACM Sigmod Record*, vol. 36, no. 2, pp. 7–12, 2007.

[30] S. Vajjala, B. Majumder, A. Gupta and H. Surana, "Text representation," in *Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems*, Sebastopol, CA, United States of America, O'Reilly Media, pp. 90–92, 2020.

[31] S. B. Kotsiantis, I. Zaharakis and P. Pintelas, "Supervised machine learning: A review of classification techniques," in *Emerging Artificial Intelligence Applications in Computer Engineering*, Amsterdam, Netherlands, IOS Press, vol. 160, no. 1, pp. 3–24, 2007.

[32] I. Muhammad and Z. Yan, "Supervised machine learning approaches: A survey," *ICTACT Journal on Soft Computing*, vol. 5, no. 3, pp. 946–952, 2015.

[33] A. Altmann, L. Toloşi, O. Sander and T. Lengauer, "Permutation importance: A corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, 2010.

[34] Scikit-learn, "4.2. Permutation feature importance," 2022. [Online]. Available: https://scikit-learn.org/stable/modules/permutation_importance.html#permutation-feature-importance