Süleyman Demirel Üniversitesi
Fen Bilimleri Enstitüsü Dergisi
Cilt 21, Sayı 1, 190-200, 2017

Süleyman Demirel University
Journal of Natural and Applied Sciences
Volume 21, Issue 1, 190-200, 2017

DOI: 10.19113/sdufbed.20964

Effects of Feature Extraction and Classification Methods on Cyberbully Detection

Esra SARAÇ[1], Selma Ayşe ÖZEL[*1]

[1]Cukurova University, Faculty of Engineering Architecture, Department of Computer Engineering, 01330, Adana

(Alınış / Received: 14.06.2016, Kabul / Accepted: 12.12.2016, Online Yayınlanma / Published Online: 30.12.2016)

Keywords
Cyberbullying,
Preprocessing,
Feature selection,
Classification

Abstract: Cyberbullying is defined as an aggressive, intentional action against a defenseless person by using the Internet, or other electronic contents. Researchers have found that many of the bullying cases have tragically ended in suicides; hence automatic detection of cyberbullying has become important. In this study we show the effects of feature extraction, feature selection, and classification methods that are used, on the performance of automatic detection of cyberbullying. To perform the experiments FormSpring.me dataset is used and the effects of preprocessing methods; several classifiers like C4.5, Naïve Bayes, kNN, and SVM; and information gain and chi square feature selection methods are investigated. Experimental results indicate that the best classification results are obtained when alphabetic tokenization, no stemming, and no stopwords removal are applied. Using feature selection also improves cyberbully detection performance. When classifiers are compared, C4.5 performs the best for the used dataset.

Sanal Zorbalık Tespitinde Nitelik Çıkarımı ve Sınıflama Yöntemlerinin Etkileri

Anahtar Kelimeler
Sanal zorbalık,
Önişleme yöntemleri,
Nitelik seçimi,
Sınıflandırma

Özet: İnternet ya da diğer elektronik içerikleri kullanarak savunmasız kişilere karşı yapılan hakaretler sanal zorbalık olarak adlandırılmaktadır. Sanal zorbalık konusunda yapılan çalışmalar, bu hakaretlerin özellikle ergen yaş grubundaki gençler için intihara kadar sonuçlanan etkilerinin olduğunu göstermektedir. Bu sebeple sanal zorbalığın otomatik tespiti oldukça önemlidir. Bu çalışmada nitelik çıkarımı, nitelik seçimi ve sınıflama yöntemlerinin otomatik sanal zorbalık tespiti üzerindeki etkileri gösterilmektedir. Deneyler FormSpring.me veri kümesi üzerinde yapılmış ve önişleme yöntemlerinin; C4.5, Naive Bayes, kNN ve SVM gibi farklı sınıflayıcıların; bilgi kazancı ve ki kare nitelik seçim yöntemlerinin etkileri araştırılmıştır. Deneysel sonuçlar, en iyi sınıflandırma performansının alfabetik karakterlerin alındığı, durma kelimelerinin silinmediği ve kelime köklerine ayırma işleminin yapılmadığı durumlarda elde edildiğini göstermiştir. Nitelik seçimi sınıflandırma performansını arttırmıştır. Kullanılan sınıflayıcılar karşılaştırıldığında C4.5, kullanılan veri kümesi için en iyi yöntem olmuştur.

1. Introduction

Cyberbullying is defined as an aggressive, intentional action against a defenseless person by using the Internet or other electronic methods such as emails, content on web sites or text messages [1, 2]. Cyberbullying contains harassment, hate, and outrage [2]. Research in cyberbullying suggests that teenagers are the main victims [3–10]. Hence, automatic detection of cyberbullying has become important for worldwide health issues among adolescents.

With the increased use of the Internet, and the ease of access to online communities provide an avenue for cybercrimes like cyberbullying. In the USA, the problem of cyberbullying has become increasingly acute, and has officially been identified as a social threat. Researchers should study cyberbullying with respect to its detection, prevention and mitigation.

Day by day, the effects of cyberbullying have become more serious for its victims [11]. In many cyberbullying cases, victims have attempted suicide due to the emotionally abusive, humiliating, and aggressive messages left by predators [12].

In the majority of cases, younger victims need to hide their predicament from adults (parents/teachers), since they think that they might loose their mobile

*Corresponding author: saozel@cu.edu.tr

phone and/or Internet access privileges if they share this situation with their elders. According to [13], female victims are more likely to report cyberbullying during early ages than male victims. Also, Özdemir [14] have shown that self-esteem of adolescents is affected negatively because of cyberbullying.

The challenges in fighting cyberbullying include: detecting online bullying when it occurs; reporting it to law enforcement agencies, Internet service providers and others; and identifying predators and their victims. In the literature, cyberbullying has been studied extensively from the social perspective, especially with respect to understanding its various attributes and its prevalence. However, very little attention has been focused on its online detection. Automated detection of cyberbullying and the provision of preventive measures are needed in fighting against cyberbullying. There exist a few studies on automatic detection of cyberbullying. The earliest work in this area belongs to Yin et al. [15] who perform harassment detection from forum and chat room datasets provided by a content analysis workshop (CAW). Contextual features are based on the similarity measure between posts, with the intuition that the posts which are dramatically different from their neighbors are more likely to be harassing posts. To detect misbehavior, a supervised learning method is used. Support vector machine classifier with content, sentiment, and contextual features of documents are applied. The characteristics of the author of the posts are not considered. Only the contents of the posts are used. N-grams, tf*idf weighting and foul words frequency are used as feature extraction and weighting. According to the experimental results, Yin et al. [15] observed that considering sentiment and contextual features improves the performance of cyberbully detection, and 61.9% recall is achieved.

The second study in this domain is Cambria et al. [16] who offer a sentiment analysis approach to detect harassment in social media. Later, Chen et al. [17] proposed the use of a lexical syntactic feature approach to detect the level of offensiveness in the comments and potentially offensive users. They also considered the writing style of the users for identification of the potential offensive users rather than for detecting bully comments.

Kontostathis et al. [18] is the first study which uses Formspring.me Web site as the data set to detect cyberbullying. The dataset used in the experiments contains 3915 posted messages crawled from the Formspring.me Web Site. Each post is labelled by Amazon Mechanical Turk. In this study, machine learning techniques like Latent Semantic Indexing and Singular Value Decomposition are used to find bullying terms. Queries are then expanded with bullying terms. An average precision of 91.25% at rank 100 is achieved. Later, the same researchers [19] devised NUM and NORM features by assigning a severity level to the bad words list obtained from nosewaring.com Web site. NUM is a count, and NORM is a normalization of the bad words, respectively. Features are grouped based on their bulliness levels as bad, worse, very bad etc. C4.5 classifier and an instance base learner, from Weka data mining tool are used for classification. The same Formspring.me dataset is used and positive examples are replicated up to ten times to balance the class distribution of the dataset. Accuracy of the proposed study is observed as up to 78.5%.

Dinakar et al. [20] labelled YouTube comments manually to develop a dataset for cyberbully detection studies. This study contains 2 steps. In the first step, the topics of comments are investigated whether comments have sensitive topics like sexuality, race/culture, intelligence, and physical attributes, or not. In the second step, the topics of the comments are determined. To do that both binary and multiclass SVM classifiers are applied. According to the experimental evaluations, it is observed that binary class classifier has better performance than multiclass classifier, and 66.7% accuracy is achieved for detecting cyberbullying.

Sanchez and Kumar [21] propose Twitter bullying detection with Naïve Bayes classifier. In this study, gender specific bullying detection is made on twitter data set, and it is obtained 67.3% accuracy values with Naïve Bayes classifier.

In another recent study on cyberbully detection [22], gender specific features are preferred and users are categorized into male and female groups. Dadvar et al. [22] have shown that taking user context, such as users' comments history and users' characteristics into account improves the performance of detection tools for cyberbullying incidents considerably. In the experiments, YouTube comments are used as the dataset and SVM is applied as the classifier.

Dadvar and Jong [23] have also shown that the accuracy of cyberbully detection increases by using the personal information of users' like gender and age. In this study an SVM classifier from Weka data mining tool is applied to a dataset from MySpace corpus. Only tf*idf values of features that are extracted from the dataset are used to eliminate infrequent terms, and no other feature selection methods are applied. For Baseline, Gender-specific, Female-specific and Male-specific approaches the F-measure values obtained are 0.20, 0.23, 0.08 and 0.28, respectively. Later, Dadvar et al. [24] have studied a YouTube data set with a multi-criteria evaluation system to clarify users' behaviors and their characteristics over expert knowledge. In this study, users have scores, which are assigned by the system, based on their previous activities. These scores show their "bulliness" level. The scores are found helpful to decide if a user is bullying or not.

In Xu et al. [25], several natural language techniques are used to detect bullying. Sentiment analysis features are used for bullying roles detection, and then topics are identified by using Latent Dirichlet Analysis. Xu et al. [25] aimed to set baselines techniques for bully detection, and invited other researchers to further study these techniques. They found seven frequent emotions, some of which have been previously well-studied, and some are non-standard in bullying. To identify these emotions, a fast training method is proposed and applied. Proposed algorithm is applied to twitter data set, which is not a conventional labeled training dataset, with SVM classifier. The overall success of this experiment reaches to 85% accuracy.

Nahar et al. [26] proposed an effective approach to detect cyberbully messages from social media through weighting schemes of feature selection. A graph model is presented to extract the cyberbullying network, which is used to identify the most active cyberbullying predators and victims through ranking algorithms. Their dataset contains data collected from three different social networks: Kongregate, Slashdot, and MySpace. Weighted tf*idf scheme is used on bullying-like features. The bad words are scaled by a factor of two, and the LDA [27] is used to generate features; and a range of top features are selected and compared to improve the classification result. LibSVM is applied to the two class classification problem using a linear kernel. For MySpace data set, they obtained 0.31 and 0.92 F-measure values for Baseline and Weighted tf*idf approaches, respectively.

Munezero et al. [28] propose a usable public dataset for harmful language detection. 98% accuracy is achieved for the proposed dataset by using NBM, SMO and J48 classifiers from Weka.

Research on online sexual predators' detection [29, 30] associates the theory of communication and text-mining methods to differentiate between predator and victim conversations, as applied to one-to-one communication such as in a chat-log dataset. For several topics related to cyberbully detection, research has been carried out based on text mining paradigms, such as identifying online sexual predators [29], vandalism detection [31], spam detection [32] and detection of internet abuse and cyber terrorism [33].

Zubiaga et al. [34] study a Twitter data set and show the positive effects of feature selection on classification performance on the Twitter data set. They presented 15 features to represent trending topics such as news, ongoing events, memes, and commemoratives. Presented features are independent of the language used in tweets. The overall success of this experiment reaches to 81.2% accuracy.

There are also some software products aimed at detecting cyberbullying, like Bsecure [35], Cyber Patrol [36], eBlaster [37], IamBigBrother [38], and Kidswatch [39]. However, these software are based on filtering methods which generally work with a simple keyword search and do not consider the semantic meaning of the text. Some filters block the Web page, if that page contains the keyword. Some filters split the actual offensive words themselves. In some products, detected keywords are removed from the page. However, this method can change the overall meaning of the sentence. Moreover, these filters can easily be skipped. Therefore, filters are not effective in avoiding cyberbullying, while there are a lot of ways to avoid inconvenient and offensive content [40]. In addition to this, users should install and maintain filtering methods manually.

The aim of this study is to show the effects of feature extraction, feature selection, and classifier used, on the performance of automated detection of cyberbullying. We experimentally investigate the effects of preprocessing methods; such as tokenization, stop word removal, stemming, and lower case conversion, as well as feature extraction, feature selection, and different classifiers on classification performance of cyberbully detection. In the literature, only a few preprocessing techniques are employed, and to our knowledge, there is no such study which investigates the effects of all preprocessing methods on cyberbully detection. We think that the results suggested here will be helpful to researchers in determining which preprocessing methods should be used for feature extraction, whether a feature selection should be used or not, and which classifier is more successful for cyberbully detection.

The rest of the paper is organized as follows: in the second section material and methods, which include preprocessing methods, dataset, feature extraction, feature selection and classification methods, used in this study are explained. In the third section experimental results are discussed; and finally section four concludes our study.

2. Material and Method

In this paper, our aim is to show the effects of all possible combinations of preprocessing techniques including tokenization, stop word removal, stemming, and lowercase conversion; as well as effects of using different portions of text for feature extraction on the accuracy of detecting cyberbullying. After that, we apply some filter based feature selection methods that are information gain and chi-square to reduce feature space so as to improve both training and testing times and accuracy of cyberbully detection. Finally, we try to determine which classifier should be used for cyberbully detection. To do that, we apply four basic classifiers that are Naïve Bayes, decision tree, support vector machines, and k

nearest neighbor to detect cyberbullying and compare their classification performances. In the below subsections the details about the material and methods used are explained.

2.1. Preprocessing methods

In this study, we experiment with four frequently used preprocessing steps that are tokenization, stopword removal, lowercase conversion, and stemming that are applied in text mining and information retrieval studies.

Tokenization is the procedure of splitting a text into tokens. These tokens can be words, phrases, or other meaningful parts. In tokenization part, tokens can be taken from only alphabetic or alphanumeric characters which are split by non-alphanumeric characters. In our study we have two cases for tokenization;

1. Alphabetic tokens which consist of only alphabetic characters.
2. Alphanumeric tokens which consist of alphanumeric characters, punctuations, and special punctuations namely emoticons.

We organize an emoticon list for the second case. The whole list of emoticons used in this study is given in Table 1. If any punctuation is in this list, we do not split it into single punctuations like ':' and ')', we consider :) as a whole.

Table 1. Emoticon List

Emoticons		
:)	(:	:D
D:	;)	(;
;(	):	:(
);	:/	/:
;D	D;	

In our study, we pay special attention to emoticons as they are used to express feelings, and we want to show that whether using emoticons as separate features improves cyberbully detection or not.

The second preprocessing step is lowercase conversion. Since the meaning of a word is not case sensitive, all uppercase characters are usually converted to their lowercase forms. However, in blogs, forums and other electronic communication platforms, uppercase characters are used for emphasizing the importance of a word, or uppercase characters mean loud speak. Therefore in our study, we have two cases for this conversion;

1. All words (terms) are converted to their lowercase forms.
2. All words (terms) except all uppercased terms are converted to their lowercase forms.

In the second preprocessing option; if all characters of a word are written in uppercase, the word stays the same, otherwise characters of the word are converted to their lowercase forms (e.g., if the original word is ABCD, it stays the same, so it is taken as ABCD. However if the original word is Abcd or abCd, it is converted into abcd).

The meaningless words on their own are called stopwords (e.g., prepositions, conjunctions, articles, etc.). Hence, in some cases such as topical classification of texts, or information retrieval stopwords have no positive effect on classification or search performance. However, cyberbully detection is different than topic classification of texts so, we study the effects of stopwords on classification performance for cyberbully detection.

The purpose of stemming is to reach root forms of derived words therefore to reduce feature space. In our study, we use Porters' stemmer [41] to show the effect of stemming.

In this study, we consider all possible combinations of the above mentioned four preprocessing methods. Tokenization is either alphabetic or alphanumeric. Lowercase conversion is either ON or OFF; that is, terms are either converted to lowercase or some of them are kept in their original forms. Stopword removal is either ON or OFF; that is, stopwords are either eliminated or kept within text. Stemming is either ON or OFF; that is, terms are either reduced to their root forms or kept in their original forms. Thus, we have 16 different preprocessing combinations. Binary codes are given to all preprocessing combinations to make them more formal and representable. Since we have 4 methods, binary code for a preprocessing combination has 4 bits like $x\,y\,z\,t$ where

$$x = \begin{cases} 0 & if\ alphabetic\ tokenization\ is\ used \\ 1 & if\ alphanumeric\ tokenization\ is\ used \end{cases} \quad (1)$$

$$y = \begin{cases} 0 & if\ lowercase\ conversion\ is\ used \\ 1 & if\ some\ features\ are\ uppercased \end{cases} \quad (2)$$

$$z = \begin{cases} 0 & if\ stemming\ is\ not\ applied \\ 1 & otherwise \end{cases} \quad (3)$$

$$t = \begin{cases} 0 & if\ stopwords\ are\ kept \\ 1 & if\ stopwords\ are\ eliminated \end{cases} \quad (4)$$

2.2. Dataset

In this study, we use Formspring.me dataset [29], which is an xml file consisting of 13124 posted messages with 50 different ids crawled from the Formspring.me Web site. An example for a post is given in Figure1. For each id, the profile information and each post (question and answer) are extracted. Each post is labeled by three workers from Amazon's

Mechanical Turk for cyberbullying content. The data contains the following xml tags:

- <BIO> is profile biography created by owner of the id,
- <DATE> is the date the id was crawled,
- <LOCATION> is the location provided by the owner of the id,
- <USERID> is the actual id itself,
- <TEXT> contains the question and answer part of the message.

```
<FORMSPRINGID>
        <BIO>Gema Loves Preston. :D</BIO>
        <DATE>20100731</DATE>
        <LOCATION>Jackson Michigan</LOCATION>
        <USERID>aguitarplayer94</USERID>
        <POST>
                <TEXT>Q: what is your favorite song?
:D A: I like too many songs to have a favorite</TEXT>
                <ASKER></ASKER>
                <LABELDATA>
                        <ANSWER>No</ANSWER>

        <CYBERBULLYWORD>n/a</CYBERBULLYWORD>
                        <SEVERITY>0</SEVERITY>
                        <OTHER></OTHER>

        <WORKTIME>13</WORKTIME>

        <WORKERID>A8PXREHJMZJPZ</WORKERID>
                </LABELDATA>
                ……
        </POST>
</FORMSPRINGID>
```

Figure 1. An example of the data set

For the label part of the data, there are 3 occurrences of <LABELDATA> tag which contains following tags:

- <ANSWER> can be either YES or NO as to whether the post contains cyberbullying or not,
- <CYBERBULLYWORD> contains word(s) or phrase(s) identified by the Mechanical Turk worker as the reason for cyberbullying (n/a or blank if no cyberbullying detected)
- <SEVERITY> gives cyberbullying severity from 0 (no bullying) to 10,
- <OTHER> includes other comments from the Mechanical Turk worker,
- <WORKTIME> is the time needed to label the post (in seconds),
- <WORKERID> is the Mechanical Turk worker id.

First of all, the data set is split into two classes as "CyberBully Positive" and "CyberBully Negative". CyberBully Positive documents contain cyberbullying, and the others do not. After this step, CyberBully Positive class includes 836 posts and CyberBully Negative class includes 12288 posts. To split the data set as train and test sets we use holdout method which is used for data sets that have similar sizes as our data set [42]. As stated in Chakrabarti [43], holdout method is applied to partition some well-known text mining benchmark datasets such as the Reuters and the 20NG which have similar feature size and number of samples as our data set. Approximately 75% of the samples in the Reuters and the 20NG data sets are randomly chosen as train set, and the rest are taken as the test set. Therefore we applied the same method to our data set. Numbers of posts in the training and test sets for both positive and negative classes are presented in Table 2.

Table 2. Train/test distribution of the dataset

	# of Posts in	
Class Label	Train	Test
CyberBully Positive	627	209
CyberBully Negative	9216	3072

2.3. Feature Extraction

In this study, features are extracted from the positive and negative posts in the training data set. To extract features we use <Text> tag in the data set. All <Text> tags have two different parts that include a question part which begins with "Q:", and an answer part which begins with "A:". We have three cases for the feature extraction step;

1. Ignore the question and answer parts of the <Text> tag, and extract the features from the whole <Text> tag. This extraction method is called as "All".
2. Use only question parts for feature extraction. This method is called as "Q".
3. Use only answer parts for feature extraction. This method is called as "A".

Numbers of features obtained according to the above feature extraction and preprocessing methods are given in Table 3. After the feature extraction step, for each feature we count document frequency which is the number of documents in the training set that contain the feature. Then, features that have a document frequency which is less than 0.1% of the number of documents in the training set are eliminated to remove misspelled words or words which are used very rarely. According to Salton [44] and our previous studies [45–47] using document frequency value of terms allows us to eliminate misspelled or unimportant terms from the feature space. The numbers of features obtained according to the three feature extraction method with a 0.1% document frequency filtering are given in Table 4.

2.4. Feature selection and classification

For each preprocessing and feature extraction method that are described above, the chi-square (CHI2) and Information Gain (IG) feature selection

processes are applied. The CHI2 test is used in statistics, among other things, to test the independence of two events [48]. In feature selection, CHI2 is used for testing whether the occurrence of a specific term and the occurrence of a specific class are independent.

Table 3. Number of extracted features

Preprocessing Code	Feature Extraction Method		
	All	A	Q
0000	14096	9250	9492
0001	13792	8959	9209
0010	11187	7469	7654
0011	11002	7270	7469
0100	15212	9679	10344
0101	14905	9386	10058
0110	12336	7912	8543
0111	12141	7704	8347
1000	21983	13534	14020
1001	21679	13243	13737
1010	13801	8687	9362
1011	13726	8598	9266
1100	23388	14083	15026
1101	23081	13790	14740
1110	16008	9693	10886
1111	15929	9600	10783

Table 4. Number of extracted features with document frequency filtering

Preprocessing Code	Feature Extraction Method		
	All	A	Q
0000	1875	1087	1089
0001	1639	876	882
0010	1768	1063	1066
0011	1578	891	899
0100	1911	1100	1124
0101	1678	887	917
0110	1816	1074	1101
0111	1618	896	930
1000	2136	1238	1186
1001	1900	1027	979
1010	1902	1169	1132
1011	1775	1039	993
1100	2161	1249	1217
1101	1922	1036	1010
1110	2027	1236	1211
1111	1896	1104	1068

Information gain (IG) computes the level of information in bits for the class prediction. IG is used if the only information available is the presence of a feature and the corresponding class distribution [49]. In IG feature selection method, an attribute with high mutual information should be preferred to other features.

In this study, CHI2 and IG are applied to training datasets to select top *n* features therefore to reduce the dataset size. After selecting the top *n* features, a classification model is learned from the training data having the selected *n* features by using the J48 (C4.5), Naïve Bayes, IBk (kNN) and SVM classifier of Weka [50] data mining tool. The *n* values are determined as 10, 50, 100, and 500 to show the effect of different feature sizes.

Classification performance is measured with F-measure [51] value. F-measure value for a class is computed as in Equation 5.

$$F - measure = \frac{2 * recall * precision}{recall + precision} \quad (5)$$

where precision is the exactness of the classification algorithm, and it is computed as in Equation 6.

$$precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (6)$$

Recall is the completeness of the classification algorithm, and it is computed as in Equation 7.

$$recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (7)$$

If dataset used in the classification does not have a balanced class distribution, computing F-measure for only one class may be misleading. As an example, we assume that we have a test dataset which has 100 instances and 90 of them belong to the negative class and 10 samples are in the positive class. Then, we assume that we apply a classifier which labels all samples as negative class in the test set. In that case, F-measure for the negative class is approximately 0.95 while F-measure for the positive class is equal to 0. Actually using any one of the F-measure value will not give the overall performance of the classifier. To overcome this problem, Macro averaged and Micro averaged F-measure values are used frequently [52]. Micro averaged F-measure is the weighted average of the F-measure by class distribution. In Macro averaged F-measure, classes have equal weights therefore, it is the arithmetic average of F-measure values computed for each class. If class distributions of the dataset is balanced, both micro and macro averages are the same. Otherwise the Macro averaged F-measure may be less than the Micro averaged F-measure. As our dataset is not balanced, we should compute either micro average or macro average of the F-measure values of the two classes. As the default average F-measure computation in Weka data mining tool is Micro averaged F-measure [50], we used this measure in our experiments. However, the experiments may be repeated for Macro averaged F-measure value as future work.

When a two-class dataset has not a balanced class distribution, and the main class of interest is represented with only a few samples while the majority of samples belong to the negative class, the dataset is said to be imbalanced [51]. To improve classifier performance with imbalanced datasets, oversampling, undersampling, threshold moving, and ensemble techniques are also used [51]. In oversampling method, positive instances are duplicated until the dataset becomes balanced. In undersampling method, randomly chosen negative

instances are removed until the dataset becomes balanced. Threshold moving pretends how the model makes decision when labelling unseen data. In the ensemble technique, a set of classifiers are combined to form a composite model. Each classifier returns a class label prediction, and then these returned predictions are combined according to weight of the classifiers. Finally, ensemble system returns the class label having the highest weight. Bagging, boosting, and random forests are examples of ensemble methods, and these methods tend to be more accurate than single classifier system. As our dataset is imbalanced one of the above methods can also be applied to improve classification performance. However, in this study our aim is to show the relative performance of the preprocessing and feature selection methods therefore we did not apply any one of the oversampling, undersampling, threshold moving, or ensemble methods. We used the dataset as it is, and we applied single classifier. As future work, one can apply one of the above methods if the aim is to develop more accurate classifier.

3. Results

During the experiments, all possible combinations of the four preprocessing tasks with three feature extraction methods are considered as mentioned before. Therefore, we have 2*2*2*2*3=48 combinations for preprocessing and feature extraction methods. After that 2 feature selection methods with 10, 50, 100, and 500 feature sizes are applied, so we have 2*4*48+48=432 experiments and we repeat all these experiments for four classifiers. First, we investigate the effects of preprocessing methods and feature extraction techniques on classification performance. To do that, we apply all classifiers for preprocessing and feature extraction combinations without making any feature selection. After that, experiments are repeated for various feature sizes such as 10, 50, 100, and 500, so that the impact of preprocessing can be comparatively observed with various feature dimensions. Detailed analyses are given in the following subsections.

3.1. Classification performance of all preprocessing and feature extraction methods

In this experiment, we show the effect of feature extraction (e.g., *All, A, Q*) and preprocessing methods on classification performance. We compare Micro averaged F-measure values of each preprocessing and feature extraction methods by using all classifiers, and the results for J48 classifier are presented in Figure 2.

According to Figure 2, for 6 preprocessing combinations "*All*" method has the maximum F-measure value; for 7 preprocessing combinations, "*A*" method has the maximum F-measure value; for 4 preprocessing combinations, "*Q*" method has the highest F-measure value. Although "*All*" and "*A*" feature extraction methods have the highest classification performance in majority of the preprocessing methods, the maximum F-measure value is obtained when "*Q*" feature extraction and 0000 preprocessing method is used among the 48 combinations; and, minimum F-measure value is obtained with "*All*" feature extraction and 1101 preprocessing method.

The best and worst preprocessing and feature extraction combinations are summarized in Table 5 for all classifiers. All results are not presented here to save space. According to Table 5, maximum F-measure values are obtained when "*Q*" feature extraction method is used for SVM, IBk and J48 classifiers.

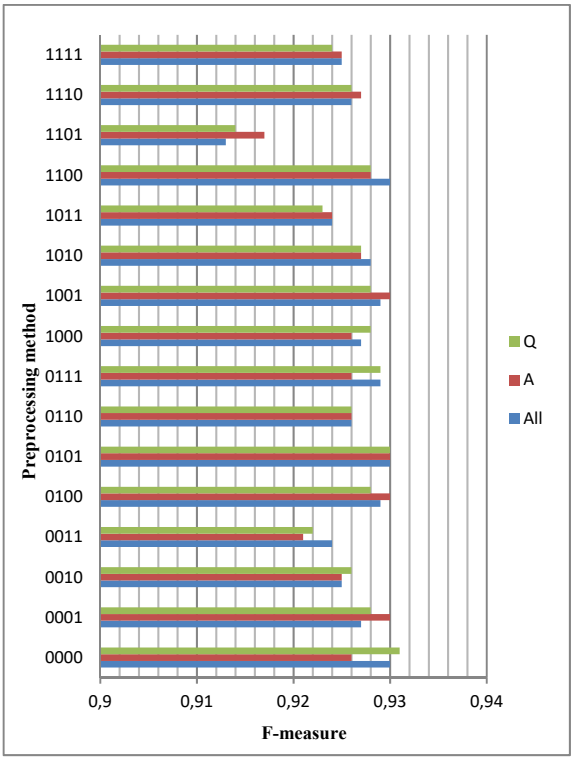

Figure 2. Comparison of classification performance without any feature selection

When we consider all preprocessing and feature extraction combinations, we can say that F-measure performance of this data set is affected positively, when alphabetic tokenization, which eliminates punctuation marks and emoticons, is used. In majority of the cases, keeping stopwords as features and not applying stemming provide better classification accuracy. Lowercase conversion is observed in majority of the best and worst cases, therefore we can conclude that terms can be converted into lowercase format to reduce feature space since using uppercase terms as separate features do not improve classification accuracy in majority of the cases. When the best and the worst F-measure values for each classifier are compared, we can also conclude that choosing right preprocessing methods improves classification accuracy 2% for J48 and SVM, 5% for NB, and 10% for IBk.

Table 5. Best and worst preprocessing and feature extraction combinations for each classifier

Classifier	Best Case	Best F-measure	Worst Case	Worst F-measure
J48	0000-Q	0.931	1101-All	0.913
NB	0001-All	0.935	1000-A	0.886
IBk	0110-Q	0.998	1011-All	0.897
SVM	0000-Q	0.927	1001-All	0.903

3.2. Feature Size Analysis

In this experiment we compare F-measure values obtained when classification is performed by making feature selection with different feature sizes. To obtain these results, we use two well-known feature selection methods that are information gain (IG) and chi-square (CHI2). A summary of the experimental results are given in Table 6 and Table 7 in which the best and worst cases for all classifiers and feature selection methods are presented.

In the best and worst case columns in Table 6 and Table 7, binary codes (i.e., the first four bits) show the preprocessing methods; after the binary code the "Q", "A" or "All" mean the feature extraction method; and the last values such as 10, 500, etc. mean the number of features used in the classification for the best and worst cases. According to Table 6 and Table 7, feature selection affects the classification performance positively for all classifiers. However, the best feature sizes are different for each classifier. The best feature sizes are 500 for J48, 10 and 50 for NB, 500 for IBk, and 10 for SVM classifiers. With the feature selection, the best F-measure performance is obtained with alphabetic tokenization, uppercase or lowercase forms, not using stems with no stop word removal. However, for NB, IBk, and SVM using all features without any feature selection yields the worst classification accuracy. For J48, the second worst classification accuracy is observed when all features (without any feature selection) are used.

Table 6. Best and worst cases for all classifiers with IG feature selection method

Classifier	Best Case	Best F-measure	Worst Case	Worst F-measure
J48	0100-Q-500	0.949	1101-All-10	0.912
NB	0000-Q-10	0.930	1000-A-All features	0.886
IBk	0100-A-500	0.997	1011-All-All features	0.897
SVM	0000-Q-10	0.923	1001-Q-500	0.903

As shown in Table 6 and 7, the performance of the two feature selection methods are similar to each other however, for NB, IBk, and SVM classifiers, chi-square feature selection gives slightly higher F-measure performance. According to our maximum F-measure values, "Q" feature extraction method gives the highest F-measure values in majority of the cases. Therefore, we can prefer "Q" feature extraction method since it yields smaller feature space with respect to "All" method. For the J48 classifier, the performances of both feature selectors are observed as the same. This result may occur due to the fact that J48 classifier chooses the best features to form the decision tree, and the classifier itself can also be used as a feature selector.

Table 7. Best and worst cases for all classifiers with CHI2 feature selection method

Classifier	Best Case	Best F-measure	Worst Case	Worst F-measure
J48	0100-Q-500	0.949	1101-All-10	0.912
NB	0001-All-50	0.935	1000-A-All features	0.886
IBk	0110-Q-500	0.998	1011-All-All features	0.897
SVM	0000-Q-10	0.927	1001-All-All features	0.903

3.3. Classifier Analysis

In this experimental task, we study the performances of J48, Naïve Bayes, IBk, and SVM classifiers. Figures 3-5 give F-measures of classification for only one of the successful preprocessing combination which is 0000 with CHI2 feature selection method for the three feature extraction techniques. Actually, the results obtained for other preprocessing combinations and IG feature selection method are similar, and to save space they are not included in this paper.

According to Figures 3–5, the IBk classifier has the best F-measure value and the SVM classifier has the worst F-measure value, this result may occur due to the fact that SVM needs some parameter optimization before using it. In the experiments we use the default parameter settings, and experiments may be repeated with parameter optimization as future work. When we choose only 10 features, classification performance of all four classifiers are similar, but when feature size is increased, accuracies of IBk and J48 increase, while classification accuracies of NB and SVM decrease. So, if IBk and J48 classifiers are used, relatively large feature size (e.g., 500) should be chosen, however if SVM and NB classifiers are used, small feature size (e.g., 10 or 50) should be preferred.

3.4. Test Time Analysis

In this section, effects of the classification algorithms and feature selection on time required to classify new instances are investigated. For this purpose, the time required to classify the unseen posts in the test data set by using the J48, Naïve Bayes, IBk and SVM classifiers, with and without feature selection, are

compared. The results of this experiment are presented in Figure 6.

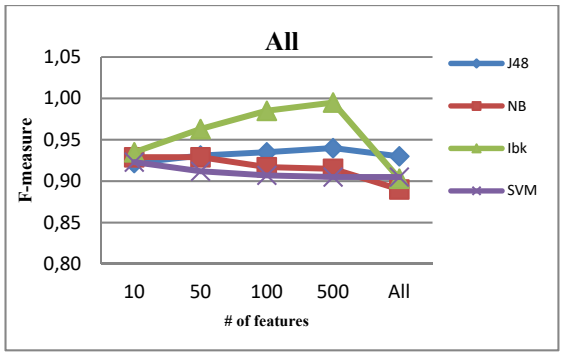

Figure 3. Classifier performances for "All" feature extraction method with CHI2 feature selection

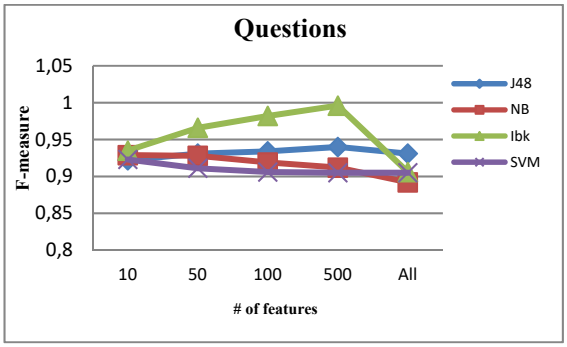

Figure 4. Classifier performances for "Q" feature extraction method with CHI2 feature selection

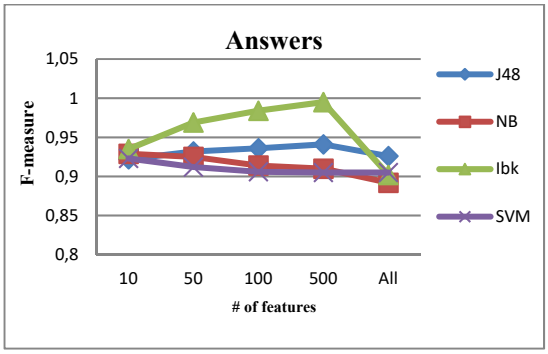

Figure 5. Classifier performances for "A" feature extraction method with CHI2 feature selection

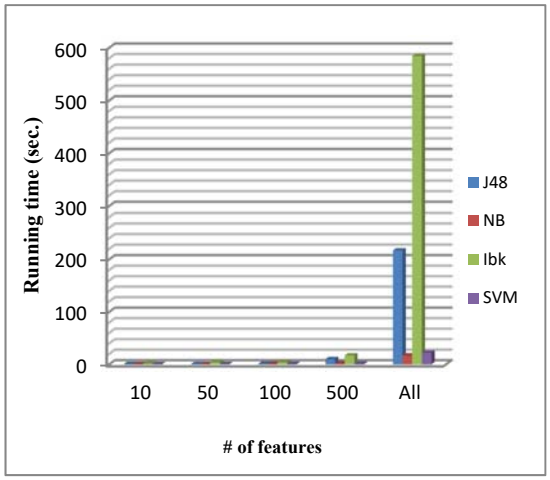

Figure 6. Comparison of run time performance of different classifiers

As it can easily be seen from the figure, making feature selection reduces the time required to classify new (unseen) data sharply, without making reduction in classification accuracy. When we compare time required by the classifiers, IBk is the slowest classifier since it is a lazy approach, then J48 is the second slowest classifier. Although NB, and SVM classifiers are fast, their classification accuracies are lower than that of IBk and J48. When feature selection is made, time required for IBk and J48 also reduces sharply. With feature selection, time required to train a classification algorithm is also reduced by similar ratios as in the testing times except for IBk since it does not require any training.

4. Discussion and Conclusion

In this paper, the impact of frequently used preprocessing tasks on text classification is empirically studied on a popular problem, cyberbully detection. We study the effects of preprocessing methods, feature extraction techniques, feature selection, and classification algorithms for detecting cyberbullying. In our experiments, for our dataset we do not observe any difference between using uppercase terms as separate features or converting all features into lowercase form in terms of classification accuracy. On the other hand, stop word removal reduces the classification accuracy. Stemming also decreases classification accuracy in most cases. We also study the effect of emoticons on cyberbullying detection. In our study we do not observe any positive effects of using emoticons as features. And lastly, we can say that, IBk and J48 classifiers are more accurate than others for this data set. Since IBk is a lazy approach, it requires too much time for classifying a new instance, and SVM requires parameter optimization before its effective usage. Hence, J48 can be used for fast and accurate classification for cyberbully detection for the used data set.

Acknowledgment

This study was supported by TUBITAK 2211-C Scholarship and Scientific Research Project Unit of Çukurova University under Grant Number: MMF2013D10.

References

[1] Snakenborg, J., Van Acker, R., and Gable, R. A. 2011. Cyberbullying: Prevention and Intervention to Protect our Children and Youth. Preventing School Failure: Alternative Education for Children and Youth, 55(2011), 88-95.

[2] Smith, P.K., Mahdavi, J., Carvalho, M., Fisher, S., Russell, S., and Tippett, N. 2008. Cyberbullying: its Nature and Impact in Secondary School Pupils. Journal of Child Psychology and Psychiatry 49(2008), 376–385.

[3] Li, Q. 2006. Cyberbullying in Schools: A Research of Gender Differences. School Psychology International, 27(2)(2006), 157-170.

[4] Agatston, P.W., Kowalski, R., and Limber, S. 2007. Students' Perspectives on Cyber Bullying. Journal of Adolescent Health 41(2007), S59–S60.

[5] Beran, T., and Li, Q. 2005. Cyber-harassment: A Study of a New Method for an old Behavior. Journal of Educational Computing Research, 32(2005), 265-277.

[6] Hinduja, S., and Patchin, J. W. 2008. Cyberbullying: An Exploratory Analysis of Factors Related to Offending and Victimization. Deviant Behavior, 29(2008), 129-156.

[7] Kowalski, R. M., and Limber, S. P. 2007. Electronic Bullying among Middle School Students. Journal of Adolescent Health, 41(6, Suppl. 1)(2007), 22-30.

[8] Ortega, R., Elipe, P., Mora-Merchán, J. A., Calmaestra, J., and Vega, E. 2009. The Emotional Impact on Victims of Traditional Bullying and Cyberbullying: A study of Spanish Adolescents. Zeitschrift Für Psychologie/Journal of Psychology, 217(4)(2009), 197-204.

[9] Patchin, J.W., and Hinduja, S. 2006. Bullies Move Beyond the Schoolyard: A Preliminary Look at Cyberbullying. Youth Violence and Juvenile Justice 4(2006), 148–169.

[10] Rivers, I., and Noret, N. 2010. "I h8 u": Findings from a Five-year Study of Text and Email Bullying. British Educational Research Journal, 36(2010), 643-671.

[11] Campbell, M.A. 2005. Cyber Bullying: An Old Problem in a New Guise? Australian Journal of Guidance and Counselling 15(2005), 68–76.

[12] What is Cyber Bullying? http://www.stopcyberbullying.org/ (Access Date: 15.12.2015)

[13] Barlett, C., and Coyne, S.M. 2014. A Meta Analysis of Sex Differences in Cyber-Bullying Behavior: The Moderating Role of Age: Sex Differences in Cyber-Bullying. Aggressive Behavior 40(2014), 474–488.

[14] Özdemir, Y. 2014. Cyber Victimization and Adolescent Self-esteem: The Role of Communication with Parents: Cyber Victimization and Self-esteem. Asian Journal of Social Psychology 17(2014), 255–263.

[15] Yin, D., Xue, Z., Hong, L., Davison, B. D., Kontostathis, A., and Edwards, L.. 2009. Detection of Harassment on Web 2.0. The Content Analysis in the WEB 2.0 (CAW2.0) Workshop at WWW2009, April 20-24, Madrid, Spain.

[16] Cambria, E., Chandra, P., Sharma, A., and Hussain A. 2010. Do not Feel the Trolls. 3rd International Workshop on Social Data on the Web (SDoW), co-located with the 9th International Semantic Web Conference (ISWC2010), Nov 7, Shanghai.

[17] Chen, Y., Zhou, Y., Zhu, S., and Xu, H. 2012. Detecting Offensive Language in Social Media to Protect Adolescent Online Safety. 2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust (SOCIALCOM-PASSAT '12), Washington, DC, USA, 71-80.

[18] Kontostathis, A., Edwards, L., and Leatherman, A. 2010. Text Mining and Cybercrime. Berry, M. W., and Kogan, J., ed. 2010. Text Mining: Applications and Theory, John Wiley and Sons, New York, NY.

[19] Reynolds, K., Kontostathis, A., and Edwards, L. 2011. Using Machine Learning to Detect Cyberbullying. 10th International Conference on Machine Learning and Applications and Workshops (ICMLA '11), December 18 - 21, Washington, DC, vol:2, 241–244.

[20] Dinakar, K., Reichart, R., and Lieberman, H. 2011. Modelling the Detection of Textual Cyberbullying. Social Mobile Web Workshop at International Conference on Weblog and Social Media, July 17-21, Barcelona, Spain.

[21] Sanchez, H., and Kumar, S. 2011. Twitter Bullying Detection. UCSC ISM245 Data Mining course report.

[22] Dadvar, M., Jong, F. D., Ordelman, R., and Trieschnigg, D. 2012. Improved Cyberbullying Detection Using Gender Information. Twelfth Dutch-Belgian Information Retrieval Workshop (DIR 2012), February 23-24, Ghent, Belgium, 23-25.

[23] Dadvar, M., and Jong F. D. 2012. Improved Cyberbullying Detection through Personal Profiles. International Conference on Cyberbullying, June 27-30, Paris, France.

[24] Dadvar, M., Trieschnigg, D., and Jong, F. D. 2013. Expert Knowledge for Automatic Detection of Bullies in Social Networks. 25th Benelux Conference on Artificial Intelligence (BNAIC), November 7-8, Delft.

[25] Xu, J., Jun, K., Zhu, X., and Bellmore, A. 2012. Learning from Bullying Traces in Social Media. Conference of North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), June 03 – 08, Montreal, Canada, 656-666.

[26] Nahar, V., Unankard, S., Li, X., and Pang, C. 2012. Sentiment Analysis for Effective Detection of Cyber Bullying. 14th Asia-Pacific International Conference on Web Technologies and

Applications (APWeb 2012), April 11-13, Kunming, China, 767-774.

[27] Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. Latent Dirichlet Allocation. Journal of Machine Learning Research, 3 (2003), 993-1022.

[28] Munezero, M., Mozgovoy, M., Kakkonen, T., Klyuev, V., and Sutinen, E. 2013. Antisocial Behavior Corpus for Harmful Language Detection. Federated Conference on Computer Science and Information Systems, September 8-11, Krakow, Poland.

[29] Kontostathis, A., Edwards, L., and Leatherman, A. 2009. ChatCoder: Toward the Tracking and Categorization of Internet Predators. Text Mining Workshop held in conjunction with the Ninth SIAM International Conference on Data Mining (SDM 2009), May 2, Sparks, NV.

[30] McGhee, I., Bayzick, J., Kontostathis, A., Edwards, L., McBride, A., and Jakubowski, E. 2011. Learning to Identify Internet Sexual Predation. International Journal of Electronic Commerce 15(2011), 103–122.

[31] Smets, K., Goethals, B., and Verdonk, B. 2008. Automatic Vandalism Detection in Wikipedia: Towards a Machine Learning Approach, Wikipedia and Artificial Intelligence: an Evolving Synergy (WikiAi08) Workshop by Association for the Advancement of Artificial Intelligence, 43–48.

[32] Tan, P. N., Chen, F., and Jain, A. 2010. Information Assurance: Detection of Web Spam Attacks in Social Media. 27th Army Science Conference, Florida.

[33] Simanjuntak, D. A., and Ipung, H. P. 2010. Text Classification Techniques Used to Facilitate Cyber Terrorism Investigation. Second International Conference on Advances in Computing, Control and Telecommunication Technologies (ACT), 198-200.

[34] Zubiaga, A., Spina, D., Martínez, R., and Fresno, V. 2015. Real-Time Classification of Twitter Trends. Journal of the Association for Information Science and Technology, 66(3) (2015), 462–473.

[35] Bsecure. http://www.safesearchkids.com/BSecure.html (Access Date: 10.05.2014)

[36] Cyber Patrol. http://www.cyberpatrol.com/cpparentalcontrols.asp (Access Date: 10.05.2014)

[37] eBlaster. Available: http://www.eblaster.com/ (Access Date: 10.05.2014)

[38] IamBigBrother. http://www.iambigbrother.com/ (Access Date: 10.05.2014)

[39] Kidswatch. http://www.kidswatch.com/ (Access Date: 10.05.2014)

[40] Butler, D., Kift, S., and Campbell, M. 2009. Cyber Bullying in Schools and the Law: Is There an Effective Means of Addressing the Power Imbalance? eLaw Journal: Murdoch University Electronic Journal of Law, 16(2009).

[41] Porter, M.F. 1980. An Algorithm for Suffix Stripping. Program, 14(3)(1980),130-137.

[42] Liu, B. 2011. Web Data Mining. Second Edition. Springer-Verlag Berlin Heidelberg.

[43] Chakrabarti, S. 2002. Mining the Web. Morgan Kaufman.

[44] Salton, G. 1968. Automatic Information Organization and Retrieval. New York: McGraw-Hill.

[45] Özel, S.A., and Saraç, E. 2011. Feature Selection for Web Page Classification Using the Intelligent Water Drop Algorithm. 2nd World Conference on Information Technology (WCIT 2011), November 23-26, Antalya, Türkiye.

[46] Saraç, E., and Özel, S.A. 2013. Web Page Classification Using Firefly Optimization. IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2013), June 19-21, Albena, Bulgaria.

[47] Saraç, E., and Özel, S.A. 2014. An Ant Colony Optimization based Feature Selection for Web Page Classification. The Scientific World Journal (2014), Article ID 649260, http://dx.doi.org/10.1155/2014/649260

[48] Yates, F., 1934. Contingency Tables Involving Small Numbers and the $\chi 2$ Test. Supplement to the Journal of the Royal Statistical Society, 1(1934), 217-235

[49] Mitchell, T. M. 1997. Machine Learning. First Edition. McGraw-Hill, New York, 432 p.

[50] WEKA. http://www.cs.waikato.ac.nz/~ml/weka/ (Access Date: 12.05.2015)

[51] Han, J., and Kamber, M. 2001. Data Mining: Concepts and Techniques (Morgan-Kaufman Series of Data Management Systems). San Diego: Academic Press.

[52] Manning, C. D., Raghavan, P., and Schütze, H., 2008. Introduction to Information Retrieval, Cambridge, UK: Cambridge University Press.