Vol.15, Issue No 2, 2025

A Natural Language Processing Powered Approach for Identify Inappropriate Language on Twitter

[1] Nallaparaju Supraja, [2] Thota Praharsha, [3] Sirra Mounika, [4] Saka Raja Sekhar,

[5] Dr. A. Rama Murthy,

[1,2,3,4,] Students, Dept. of CSE, DNR College of Engineering & Technology, Balusumudi, Bhimavaram, India.

[5] Professor, Dept. of CSE, DNR College of Engineering & Technology, Balusumudi, Bhimavaram, India.

ABSTRACT

More and more "cyber" conflicts are breaking out between individuals of diverse cultural and psychological backgrounds as a result of the increased directness of communication made possible by the proliferation of social networks and microblogging websites. This has led to an alarming increase in the prevalence of hate speech in these public forums. Whether it's based on gender (sexism), race (racism), or religious beliefs and practices, hate speech is the use of hostile, violent, or insulting language directed towards a specific group of people who share a common property. Even though the vast majority of social media and microblogging platforms have anti-hate speech policies in place, it is practically hard to police all of their content due to their massive user bases. As a result, there is a pressing need to automatically identify hate speech and filter out any content that contains it or language that incites hatred. In this study, we provide a method for identifying hate speech on Twitter. Our method relies on patterns and unigrams automatically extracted from the training data set. A machine learning algorithm is trained using these patterns and unigrams, among other features. Using a test set of tweets from 2010, we found that our method achieved an accuracy of 87.4% in binary classification and 78.4% in ternary classification when asked to determine if a tweet was offensive, hateful, or clean. Index Terms: Twitter, hate speech, sentiment analysis, machine learning.

INTRODUCTION

More people are visiting microblogging and online social networks (OSN) than any other kind of website. Platforms like Instagram, Facebook, and Twitter are attracting users from all walks of life and all corners of the globe. Their ever-expanding data sets provide an intriguing illustration of the so-called big data phenomenon. Researchers interested in automated analysis of people's views and the structure/distribution of users in networks, etc., have been drawn to big data. It is nearly hard to control the content of these websites due to their nature and the large number of posts, comments, and messages exchanged, even though they provide a platform for people to discuss and share ideas and opinions. On top of that, many individuals resort to hostile and abusive rhetoric when interacting with those who do not share their origin, culture, or beliefs. According to King and Sutton [1], 58% of the 481 anti-Islamic hate crimes that happened in the year after 9/11 happened within two weeks of the event. But these days, more and more conflicts are breaking out after every major event, thanks to the exponential rise of OSN. Unfortunately, such language is still present in OSN, even though content censorship is still a divisive issue with two camps of opinion [2]. Compared to other "cleaner" speeches, it spreads even more easily among both young and old. This is why Burnap and Williams [3] argued that decision-makers can study the spike in hate crimes after "trig-ger" events by gathering and analyzing data over time. Since many hate crimes go unreported, there is a dearth of "of_cial" data pertaining to these incidents. In this regard, social media platforms offer a more complete and accurate picture, but one that is also less trustworthy and rife with noise. We offer an effective method to identify offensive tweets and hate speech in this work to circumvent this noise and the unreliability of the data. Our method for detection is based on emotive traits, writing patterns, and unigrams. The remainder of this paper is structured as follows: in Section II we present our motivations and describe some of the related work. In Section III we officially de_ne the purpose of our study and discuss in detail our proposed approach for hate speech identification and howfeatures are extracted. Our experimental findings are detailed and discussed in Section IV. This report

closes in Section V, which also suggests potential avenues for further research.

MOTIVATIONS AND RELATED WORK

A. MOTIVATIONS

Hate speech is a specific kind of offensive language in which the speaker expresses their opinions based on their segregationist, racist, extreme, or stereotypical backgrounds. Speech that "expresses hatred of a particular group of people" is defined by Merriam-Webster as hate speech. "Speech that is intended to insult, offend, or intimidate a person because of some trait (as race, religion, sexual orientation, national origin, or disability)" is how it is defined according to the law. For this reason, numerous nations and groups have started taking a stance against hate speech as a global issue. Since people's interactions are now more indirect and their speech is more aggressive when they feel physically safe, this problem has only gotten worse with the expansion of the internet and the proliferation of online social networks. What's more, many hate groups view the internet as a "unprecedented means of communication of recruiting." [2]. Hate speech on the internet and social media not only exacerbates existing tensions but also has the potential to damage companies and spark major conflicts in the real world. Social media platforms like Facebook, YouTube, and Twitter have taken a stand against hate speech. It is always challenging to regulate and filter all the material, nevertheless. Consequently, there have been studies in the field that aim to automatically identify hate speech. Creating dictionaries of hate terms and expressions [4] or binary classifying into "hate" and "non-hate" [5] are common aims of these hate speech detection efforts. Nevertheless, determining whether a phrase includes hate speech may be challenging, especially when the speaker is attempting to hide their hate speech behind sarcasm or when there are no explicit terms that demonstrate racism, stereotyping, or hatred. On top of that, OSN are rife with comedic and sarcastic humor that, at first glance, might come off as racist, sexist, or otherwise inappropriate. The two tweets that follow provide an example: Hey there, dude. We wanted a woman's perspective, so we'll ask you, dear..., because it's been a long since we last read one of your pointless remarks. To begin with, remain silent. In its first form, the tweet seems insulting and degrading to its intended recipient. The tweet is really a joke between two pals, however, since both accounts follow each other. Similarly, in the second case, the user seems to be insulting women, although the tweet was not intended to insult women or even the intended recipient—it was part of a smaller conversation between a group of friends. This kind of language, along with others that make specific references to gender, color, ethnicity, or religion, is often used in a humorous setting, but it must be clearly differentiated from hate speeches. Dictionaries and n-grams in general may not be the best choice for identifying hateful statements, so to speak. One may make the case that sentiment analysis methods may be used to identify hate speech. But this is a distinct job that calls for higher-level tools: Returning to the original concept of identifying any existing good or negative phrase or expression, the primary objective of sentiment analysis is to determine the sentiment polarity of the tweet. With few exceptions (the word "bad" cannot be read positively under all circumstances), it is simple to depend on the literal meaning of words since words often have the same emotive polarity regardless of context or true meaning. Some words in hate speech may be bad or even carry the sense of hatred, but they are not always hate speech because of the context. The following two instances illustrate the point: Every time they lose, it annoys me. "It's simply unfair!" Although the term "hate" is used in this context, it does not constitute hate speech since the individual being insulted is not being targeted based on his gender, race, or any other protected characteristic. - A "I despise these neggers; they never stop causing me misery": Clearly, this is an expression of bigotry directed at a particular ethnic community. Because of its context dependence and the fact that we shouldn't depend on simple words or even n-grams to identify it, hate speech identification is a very different and more difficult problem than sentiment analysis. Similar to other text classification tasks, writing patterns have been successful in sarcasm detection[6,7], multi-class sentiment analysis[8,9], and sentiment quanti_cation. The application dictates the pattern types, as well as their construction and extraction methods. Hence, in this study, we use a pragmatic technique to identify patterns of hate speech and offensive texts; these patterns, together with other aspects, will be used to identify hate speech in Twitter's short text messages. Hence, we provide several sets of features in this study, such as hate speech unigrams and writing patterns. We classify tweets—texts retrieved from Twitter—into three categories, which we call "Clean," "Offensive," and "Hateful," by combining these attributes. In what follows, we'll go into more detail about each category. Here are the key points of this paper: 1) To identify hate speech on Twitter, we

suggest a pattern-based method. To do this, we define a set of parameters to maximize pattern collection and extract patterns pragmatically from the training set. 2) Along with patterns, we provide a method that pragmatically gathers hateful and offensive words and expressions and uses them in conjunction with patterns and other sentiment-based characteristics to identify hate speech. 3) Future efforts on hate speech identification may employ the suggested collections of unigrams and patterns as pre-built dictionaries. 4) Instead of merely labeling tweets as disrespectful or displaying hatred, we now divide them into three separate categories. B. Tasks Connected Many fields have made extensive use of OSN subjective language analysis, including sentiment analysis[10]_[12], sarcasm detection[6,7], rumor detection[13], and many more besides. Hate speech identification, however, has received much less attention in the literature than the subjects listed before. A few of these studies, including those by Warner and Hirschberg [5] and Djuric et al. [14], focused on phrases found on the internet. In the binary classification challenge, the first effort had a classification accuracy of 94% and an F1 score of 63.75%, whereas the second work achieved an accuracy of 80%. Extracting phrases from many prominent "hate sites" in the US was done by Gitari et al. [15]. "Strongly hateful," "weakly hateful," and "non-hateful" were the three categories into which they placed each phrase. Using grammatical pattern features and semantic feature characteristics, they trained the classifier on a test set and achieved an F1-score of 65.12%. Nobata et al. [16] achieved a 90% accuracy rate in their classification test using a combination of lexical features, n-gram features, linguistic features, syntactic features, pretrained features, "word2vec" features, and "comment2vec" features. However, there were other studies that focused on finding hate speech on Twitter. One area that Kwok and Wang [17] focused on was finding racist tweets directed toward Black individuals. They achieved a 76% success rate in binary classification using unigram features. The gathered unigrams are obviously associated to the targeted group if the hate speech is directed against a certain gender, ethnicity, race, or other. Because of this, the constructed unigram dictionary is not applicable to the detection of hate speech directed against other groups. To differentiate between hate speech and clean speech, Burnap and Ohtsuki [3] used typed dependencies, which are the relationships between words, in conjunction with bag of words (BoW) characteristics.

PROPOSED APPROACH

The objective of this work is to sort all Tweets into one of three categories, given a batch of them: _ Clean: This category includes tweets that are free of hate speech, are neutral, and do not include any objectionable language. The tweets in this category are offensive, but they do not express hatred or make racial or segregationist statements. The term "hateful" refers to tweets that are offensive and include racist, sexist, or otherwise discriminatory language. To do the classification, we use machine learning techniques: after parsing each tweet for a collection of characteristics, we apply the features to a training set and then run the classification. Part A. Information We have gathered and merged three separate data sets for this project: The first dataset to be made accessible to the public on CrowdFlower2 includes over 14,000 tweets that have been carefully categorized as either "Hateful," "Offensive," or "Clean." Three individuals have manually annotated all of the tweets in this dataset. _ Additionally, Crowd-ower3 offers a second publicly accessible data set that has been utilized in [19] and has been manually annotated into one of the three classes: "Hateful," "Offensive," and "Nei-ther," with the latter one alluding to the "Clean" class that was previously indicated. There is a third data set that was used in the study and is available on github4: There are three categories into which tweets in this dataset are sorted: "Sexism," "Racism," and "Neither." We have included the first two classes—"Sexism" and "Racism"—in the "Hateful" class because they pertain to specific types of hate speech; however, we have discarded the tweets from the "Neither" class because it is unclear whether they are clean or offensive. We manually checked a number of tweets and found that they belonged to both classes. As mentioned earlier, the three data sets were amalgamated into a larger dataset, which we subsequently partitioned as shown below. The data set is divided into three subgroups in order to carry out the classification task: A training set: this set includes 20,000 tweets, equally split between the three categories "Clean," "Offensive," and "Hateful," with 7,000 tweets in each category. Going forward, this collection will be called the "training set" for short. There are 2,010 tweets in the test set, with 670 tweets for each class. The purpose of this collection, which we will call the "test set," is to fine-tune our strategy. _ A set for validation: this set includes 2,010 tweets, with 670 tweets for each class. We will review this collection, which we will call the "valication set," to see how well our new method works. Using an equal amount of tweets for each batch ensures a fair outcome. To account for the fact that the "Hateful" class had the fewest tweets (8,340),

we decided to use 7,000 tweets for training, 670 for testing, and 670 for validation across all three classes.

DATA PRE-PROCESSING

We brie_y detail the preprocessing steps for the tweets here. The several processes carried out at this stage are shown in Fig 1.

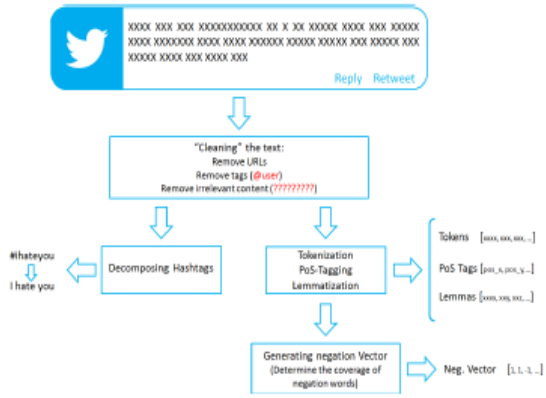

FIGURE 1. Pre-processing phases of the tweets.

The first thing we do is tidy up the tweets. This involves eliminating unnecessary expressions (words written in languages not permitted by ANSI code), tags (such as "@user"), and URLs (which start with either "http://" or "https://"). The reason for this is because they do not provide any more information on the possibility of hate speech in the tweet. In the case of tags in particular, knowing the connection between the tweet's author and the tagged person might be useful. Nevertheless, we do not find the use of tags to be beneficial to our work since no context is provided between the author and the tagged individual.

FEATURES EXTRACTION

While sentiment analysis and polarity recognition are quite different tasks, it is nonetheless reasonable to employ sentiment-based characteristics as the foundational features for hate speech detection. This is due to the fact that "negative" tweets are more likely to include hate speech than “positive" ones. For this reason, we begin by collecting data that can tell us whether a tweet is favorable, bad, or neutral. Although it is an additional step toward the primary goal of this work—the identification of hate speech—the detection of polarity is not its intended use. Hence, we extract the following characteristics from each tweet:

- the total score of positive words (PW),
- the total score of negative words (NW),
- the ratio of emotional (positive and negative) words $\rho(t)$ defined as: $\rho(t) = \frac{PW-NW}{PW+NW}$; $\rho(t)$ is set to 0 if the tweet has no emotional words,
- the number of positive slang words,
- the number of negative slang words,
- the number of positive emoticons,
- the number of negative emoticons,
- the number of positive hashtags,
- the number of negative hashtags.

The overall score of positive and negative phrases is determined using SentiStrength, a technique that assigns sentiment values to both the sentences and the individual words that make them up. Words with negative connotations have ratings between -5 and -1, whereas positive words have values between 1 and 5. Assuming a tweet t has positive polarity, we can begin by adding up the scores of individual words and assigning that total to the first set of features. Then, for negative polarity, we repeat the process and assign the absolute value of the sum to the second set of features. In this case, both sets of features take positive values.

- the number of exclamation marks,
- the number of question marks,
- the number of full stop marks,
- the number of all-capitalized words,
- the number of quotes,
- the number of interjections,
- the number of laughing expressions,
- the number of words in the tweet.

An independent characteristic that may be either "true" or "false" is a unigram, and these unigrams are pragmatically culled from the training set. We take every unigram from the training set that has a part-of-speech (PoS) tag for a noun, verb, adjective, or adverb and put it in one of three lists, one for each class. We also keep track of how many times it appears in each category. To ensure that only words with a minimum of minu occ occurrences are retained, we set this threshold as the minimum number of unigrams that must be considered.

$$\rho_{12}(w) = \frac{N_1(w)}{N_2(w)} \quad (1)$$

$$\rho_{13}(w) = \frac{N_1(w)}{N_3(w)} \quad (2)$$

where the word's frequency in class i is represented by Ni(w). The value is set to 2 if the ratio's

denominator is 0. All terms in those three classes that meet the aforementioned occurrence criteria are treated in this way. A second criterion is defined as follows, and we only retain terms that meet it:

$$\rho_{ij}(w) \geq Th_u \quad (3)$$

in which Thu is a ratio threshold that must be fine-tuned for optimal accuracy.
In each tweet, we determine whether the word w is used or not; this is in keeping with the previous statement that each of the generated words will serve as a distinct feature. "True" is the value assigned to the relevant feature if the tweet includes the term; otherwise, "false" is the value set. Figure 2 shows the most common top words extracted from the class tweets, "hateful," and Figure 3 shows the most common top words extracted from the class tweets, "offensive," given the optimal values of the two parameters minu occ and Thu (we will describe the optimization process of the different parameters later in this section).

FIGURE 2. Hateful class top words.

Words from these types are often used to insult or degrade others, but some of them have racist or sexist content or refer to certain genders, ethnic groups, or others (e.g., "muslims," "islamic," "faggot," "spic," etc.). Using a larger training set, we think we can use the method we suggested for "unigram-features" to construct

FIGURE 3. Offensive class top words.

a lexicon of terms connected to bigotry for potential use in other publications.
We were able to extract 1,373 words in total. The result is 1,373 unique unigram characteristics.

PATTERN FEATURES

Similar to how unigrams are extracted, pattern features are also taken from the training set. However, before we explain how the values of pattern features are assigned, we must first create a pattern in our environment. As a first stage, we classify tweet terms into two groups: "SW" (meaning "sentimental word") and "NSW" (meaning "non-sentimental word"). This is done on the basis of whether or not the words in the tweet have the potential to evoke strong emotions. Just like any other word, nouns, verbs, adjectives, and adverbs may evoke strong emotions. So, "SW" is any word in the tweet with a PoS that describes a noun, verb, adjective, or adverb. A term is considered to be part of "NSW" if it has another PoS tag. Following the steps outlined in TABLE 1, a pattern may be retrieved from a tweet by replacing each word that corresponds to "SW" with its simplified PoS tag and polarity. For instance, the term "Negative_ADJECTIVE" will substitute for the word "coward." The simplified PoS tag is used in lieu of the term if it belongs to "NSW," as shown in TABLE 1.

TABLE 1. List of PoS tags and their corresponding simplified tags.

PoS Tags	Simplified Tags
NN, NNS, NNP, NNPS	NOUN
VB, VBD, VBG, VBN, VBP, VBZ	VERB
RB, RBR, RBS	ADVERB
JJ, JJR, JJS	ADJECTIVE
CC	COORDCONJUNCTION
CD	CARDINAL
DT	DETERMINER
EX	EXISTTHERE
FW	FOREIGNWORD
IN	PREPOSITION
LS	LISTMARKER
MD	MODAL
PDT	PREDETERMINER
POS	POSSESSIVEEND
PRP, PRP$	PRONOUN
RP	PARTICLE
SYM	SYMBOL
TO	TO
UH	INTERJECTION
WDT, WP, WP$, WRB	WHDETERMINER
Punctuation marks	.

We take the training set and use it to extract various patterns, which we then store in three separate lists along with the frequency with which they appear. If they don't seem to be more than minimum occupancy, we remove them. Next, we have a pattern p that was in one of the three lists (let's call it C1), and we take two ratios characterized as follows:

$$\rho_{12}(p) = \frac{N_1(p)}{N_2(p)} \quad (4)$$

$$\rho_{13}(p) = \frac{N_1(p)}{N_3(p)} \quad (5)$$

in where Ni(p) is the count of pattern p instances in class i. The value is set to 2 if the ratio's denominator is 0. Solely designs meeting the criteria

$$\rho_{ij}(p) \geq Th_p \quad (6)$$

are maintained, with Thp serving as a threshold that we customize.
The total number of features retrieved from the patterns is 1,875 using the best possible values for the parameters minp occ and Thp.
For each given pattern p, we assign a numerical value to the relevant characteristic that indicates how similar the tweet is to that pattern. Accordingly, we establish the following similarity function [6] for a pair of tweets t and patterns p:

$$res(p,t) = \begin{cases} 1, & \text{if the pattern appears in the tweets as it is,} \\ \alpha \cdot n/N, & \text{if the tweet contains } n \text{ out of the } N \text{ tags of the pattern in the correct order,} \\ 0, & \text{if the tweet doesn't contain any of the tags of the pattern.} \end{cases}$$

where α is a parameter to optimize.

PARAMETERS OPTIMIZATION

In order to achieve the highest possible classification accuracy, it is necessary to optimize the many parameters presented by the suggested feature sets. Here are the parameters that need to be optimized:

- the minimal occurrence of words min_{occ}^{u}
- the word ratios threshold Th_u
- the minimal occurrence of patterns min_{occ}^{p}
- the pattern ratios threshold Th_p
- the pattern length L
- the coefficient α

We find the ideal value for one parameter at a time by eXchanging all the others. Hence, we do the following parameter settings in order to get the optimal value of the minu occ parameter:

- $Th_u = Th_p = 1.4$,
- $min_{occ}^{p} = 3$,
- $L = 7$
- $\alpha = 0.1$

We conducted our trials on each feature family separately, utilizing the values of comparable parameters established in prior work [6], to try to narrow the intervals of the parameter values. This led us to the values that were ultimately chosen. To get the present values, we next tweaked the characteristics.
To do this, we experiment with various values of the occ parameter. Figure 4 displays the outcomes. For minimu occ D 9, the best result was achieved.

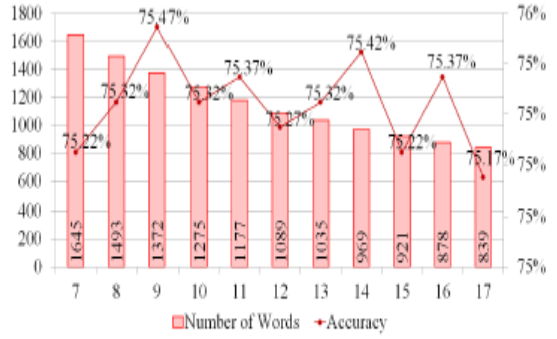

FIGURE 4. Classification accuracy (right axis) and number of words collected (left axis) for different values of the parameter min^{u}_{occ}.

After that, we stick with the current parameter settings, change Thu to 9, and set minu occ to 9. After testing values between 1.1 and 2, the best one for Thu D 1:4 was found (see Fig. 5). A grand total of 1,373 words have been compiled. We attempt several values of the parameter L and adjust the values of the parameters linked to unigram features to their ideal levels to identify the best length of patterns (L), as shown in Figure 6. All of the other parameters remained unchanged from their original settings. A total of 1,875 designs were obtained, and the ideal value for L D 5 was determined.

FIGURE 5. Classification accuracy (right axis) and number of words collected (left axis) for different values of the parameter Th_u.

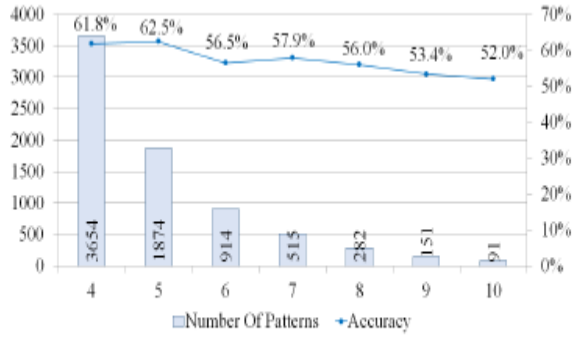

FIGURE 6. Classification accuracy (right axis) and number of patterns collected (left axis) for different values of the parameter *L*.

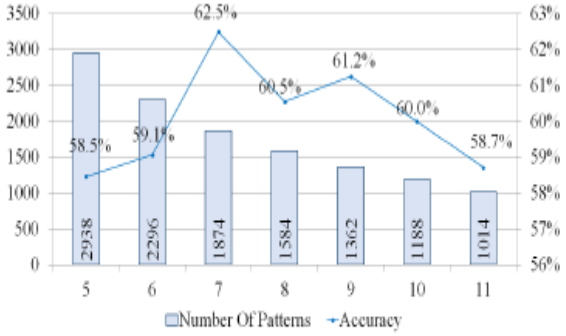

FIGURE 7. Classification accuracy (right axis) and number of patterns collected (left axis) for different values of the parameter min^{p}_{occ}.

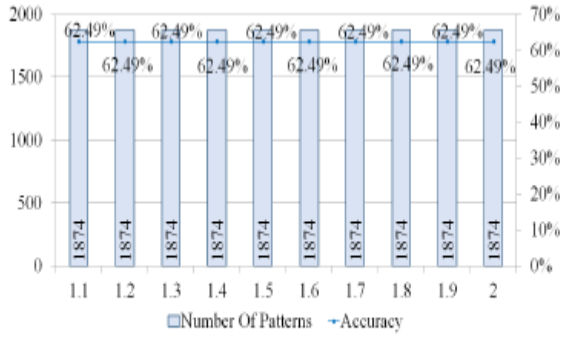

FIGURE 8. Classification accuracy (right axis) and number of patterns collected (left axis) for different values of the parameter Th_p.

Finally, we experimented with various values of and adjusted all four parameters to their ideal levels. The findings that were obtained were rather similar, bearing in mind that they should have a low value. This parameter's ideal value is equal to down to the 0.1 degree.Hence, moving on with this project, we took into account the first this scenario while preserving the following parameter values:

$$\begin{cases} min_{occ}^{u} = 9, \\ Th_u = 1.4, \\ min_{occ}^{p} = 7, \\ Th_p = 1.4, \\ L = 5, \\ \alpha = 0.01. \end{cases}$$

EXPERIMENTAL RESULTS

We go on to the final trials after feature extraction and parameter tuning. The classification is carried out by use of the Weka toolbox [22]. Different types of classifiers (e.g., rule-based, decision tree-based, etc.) are presented by Weka. The four different key performance indicators (KPIs) used to assess classification performance are recall, accuracy, percentage of true positives, and the F1-score, which is defined as:

$$F1\text{-}score = 2 \times \frac{Precision \cdot Recall}{Precision + Recall} \qquad (7)$$

Our approach relies on the machine learning algorithm "J48graft" [23] to carry out the classification. The confidence threshold for pruning (C) is the primary tuning parameter in the "J48graft" method. During this investigation, the optimal value of this parameter was found to be C D 0:04. This classifier outperforms all others, even robust ones like Random Forest and Support Vector Machine (SVM), which is why it is used in this context. There are hundreds of binary features in "J48graft" that can only take on the values "true" or "false," which may explain why it performs better than SVM when dealing with numerical data. But this still doesn't address the question of why "J48graft" is better than Random Forest.

TABLE 2. Accuracy, precision, recall and F1-score of classification using different classifiers.

	TP Rate	FP Rate	Prec.	Recall	F1-Score
Rand. Forest	0.592	0.204	0.605	0.592	0.589
SVM	0.57	0.276	0.643	0.57	0.599
J48graft	0.784	0.108	0.793	0.784	0.784

Table 2 compares the classification performances of "J48graft" with those of other classifiers. To begin, we optimize the features' settings on the test set, and then we run the classification on that set. This is to ensure that the parameters of the classifier being used, namely "J48graft," are optimized as well. We re-run the classification on the validation set after the settings are improved. So that we know the features and classifier parameters aren't leaking into our current test set, we may compare their performance to that of a separate set.

BINARY CLASSIFICATION

We began by merging all tweets labeled as "offensive" with those labeled as "hateful," as the former certainly includes aggressive and insulting language. The goal is to transform the classification into a binary classification problem. There are 14,000 tweets labeled as "offensive" and 7,000 labeled as "clean" in the training set. On the other hand, there are 1,340 tweets in the "clean" category and 2,680 in the "offen-sive" category in the test set. Execute the classification using these sets. Table 3 displays the outcomes, and Table 4 displays the confusion matrix.

TABLE 3. Binary classification performances on the test set.

Class	TP Rate	FP Rate	Prec.	Recall	F1
Sentiment-based Features					
Offensive	0.963	0.767	0.715	0.963	0.821
Clean	0.233	0.037	0.757	0.233	0.356
Overall	0.719	0.524	0.729	0.719	0.666
Semantic Features					
Offensive	0.984	0.976	0.669	0.984	0.796
Clean	0.024	0.016	0.432	0.024	0.045
Overall	0.664	0.656	0.590	0.664	0.554
Unigram Features					
Offensive	0.778	0.091	0.945	0.778	0.853
Clean	0.909	0.222	0.671	0.909	0.772
Overall	0.821	0.135	0.8534	0.821	0.826
Pattern Features					
Offensive	0.690	0.284	0.830	0.690	0.754
Clean	0.716	0.310	0.536	0.716	0.613
Overall	0.699	0.292	0.732	0.699	0.707
All features combined					
Offensive	0.876	0.122	0.935	0.876	0.904
Clean	0.878	0.124	0.780	0.878	0.826
Overall	**0.877**	**0.123**	**0.883**	**0.877**	**0.878**

TABLE 4. Binary classification confusion matrix.

Class	Classified as	
	Offensive	Clean
Offensive	**1174**	166
Clean	82	**588**

The validation set, which has not been touched by any optimization process steps—is subsequently subjected to the binary classification. Table 6 contains the confusion matrix, while Table 5 has the classification results. Using all the characteristics together yields an accuracy of 87.4 percent and a precision of 93.2 percent for the "offensive" class. Results further down by feature family reveal that

pattern features (at 70% accuracy) and unigram features (82.1%) are the most effective. The pragmatic approach to feature extraction resulted in characteristics that were strongly associated with the various classifications. That is to say, although marks based on punctuation and sentiment have not been picked out to reflect anything in particular and have been taken straight from the various tweets, patterns and top words are polarized features, and the presence of any of them greatly impacts the determination of whether a tweet is offensive or not.

TABLE 5. Binary classification performances on the validation set.

Class	TP Rate	FP Rate	Prec.	Recall	F1
Sentiment-based Features					
Offensive	0.963	0.767	0.715	0.963	0.821
Clean	0.233	0.037	0.757	0.233	0.356
Overall	0.719	0.524	0.729	0.719	0.666
Semantic Features					
Offensive	0.989	0.976	0.669	0.984	0.796
Clean	0.024	0.016	0.432	0.024	0.045
Overall	0.664	0.656	0.590	0.664	0.554
Unigram Features					
Offensive	0.778	0.091	0.945	0.778	0.853
Clean	0.909	0.222	0.671	0.909	0.772
Overall	0.821	0.135	0.854	0.821	0.826
Pattern Features					
Offensive	0.710	0.321	0.816	0.710	0.759
Clean	0.679	0.290	0.539	0.679	0.601
Overall	0.700	0.311	0.723	0.700	0.706
All features combined					
Offensive	0.875	0.128	0.932	0.875	0.902
Clean	0.872	0.125	0.777	0.872	0.821
Overall	**0.874**	**0.127**	**0.88**	**0.874**	**0.875**

TABLE 6. Binary classification confusion matrix of the validation set.

Class	Classified as	
	Offensive	Clean
Offensive	**1172**	168
Clean	86	**584**

The classification accuracy of semantic characteristics is low, however. This is due to the fact that these characteristics cannot detect hate speech, inflammatory content, or clean language when used alone. Put another way, these qualities aren't useful on their own; they need the other sets of features for proper interpretation. When it comes to sentiment-based features, it's important to consider more than just the tweet's positivity or negativity when making judgments about the content and language used. For example, inflammatory language is more often seen in negative tweets.

TERNARY CLASSIFICATION

Table 7 shows that the test set classification has much poorer accuracy, precision, and recall. By dividing the original "offensive" class into two subclasses—"offensive" and "hate-ful"—the classification accuracy drops about 10%, reaching 79.7 percent. As seen in TABLE 8, the tweets belonging to these two categories are often mistaken for one another due to their similar content, which results in reduced accuracy and recall when compared to the "clean" category.

TABLE 7. Ternary classification performances on the test set.

Class	TP Rate	FP Rate	Prec.	Recall	F1
Sentiment-based Features					
Hateful	0.390	0.216	0.474	0.390	0.428
Offensive	0.363	0.161	0.529	0.363	0.655
Clean	0.696	0.399	0.466	0.696	0.671
Overall	0.483	0.259	0.490	0.483	0.648
Semantic Features					
Hateful	0.219	0.227	0.326	0.219	0.262
Offensive	0.655	0.509	0.392	0.655	0.490
Clean	0.284	0.185	0.434	0.284	0.343
Overall	0.386	0.307	0.384	0.386	0.365
Unigram Features					
Hateful	0.639	0.058	0.846	0.639	0.728
Offensive	0.701	0.045	0.887	0.701	0.783
Clean	0.931	0.261	0.641	0.931	0.759
Overall	0.757	0.121	0.791	0.757	0.757
Pattern Features					
Hateful	0.281	0.117	0.545	0.281	0.370
Offensive	0.734	0.047	0.886	0.734	0.803
Clean	0.858	0.399	0.518	0.858	0.646
Overall	0.624	0.188	0.650	0.624	0.726
All features combined					
Hateful	0.685	0.1	0.774	0.685	0.727
Offensive	0.793	0.036	0.917	0.793	0.850
Clean	0.913	0.169	0.730	0.913	0.812
Overall	**0.797**	**0.101**	**0.807**	**0.797**	**0.796**

TABLE 8. Ternary classification confusion matrix of the test set.

Class	Classified as		
	Hateful	Offensive	Clean
Hateful	**459**	41	170
Offensive	83	**531**	56
Clean	51	7	**612**

TABLE 9. Ternary classification performances on the validation set.

Class	TP Rate	FP Rate	Prec.	Recall	F1
Sentiment-based Features					
Hateful	0.337	0.205	0.451	0.337	0.386
Offensive	0.394	0.182	0.520	0.394	0.448
Clean	0.664	0.415	0.445	0.664	0.533
Overall	0.465	0.267	0.472	0.465	0.456
Semantic Features					
Hateful	0.233	0.232	0.334	0.233	0.274
Offensive	0.634	0.467	0.404	0.634	0.494
Clean	0.300	0.217	0.409	0.300	0.346
Overall	0.389	0.305	0.382	0.389	0.371
Unigram Features					
Hateful	0.636	0.073	0.813	0.636	0.714
Offensive	0.652	0.050	0.867	0.652	0.744
Clean	0.924	0.271	0.630	0.924	0.749
Overall	0.737	0.131	0.770	0.737	0.736
Pattern Features					
Hateful	0.328	0.114	0.590	0.328	0.422
Offensive	0.721	0.053	0.872	0.721	0.789
Clean	0.845	0.386	0.523	0.845	0.646
Overall	0.631	0.184	0.661	0.631	0.619
All features combined					
Hateful	0.699	0.104	0.770	0.699	0.732
Offensive	0.763	0.048	0.889	0.763	0.821
Clean	0.891	0.172	0.722	0.891	0.798
Overall	**0.784**	**0.108**	**0.793**	**0.784**	**0.784**

the categorization, and distinguish between insulting and hateful messages. As we saw in the justification, hate speech often targets whole groups of individuals because of their histories, but an insulting letter may only target one

TABLE 10. Ternary classification confusion matrix of the validation set.

Class	Classified as		
	Hateful	Offensive	Clean
Hateful	**468**	48	154
Offensive	83	**511**	76
Clean	57	16	**597**

recipient of the communication. It is sometimes quite difficult to determine whether a tweet is nasty or just insulting, even for those who do not know the speaker. We conducted the classification using identical sets of features. You can see the results of the classification in TABLE 9. Because it is difficult to differentiate between hostile and offensive statements, the accuracy fell dramatically when compared to binary classification. As seen in TABLE 10, a few of the "hateful" tweets were incorrectly categorized as "clean." This explains why the "hateful" class has a poor recall and why the "clean" class has a poor accuracy. The reason for this is because there are certain "hateful" and "clean" tweets that cannot be distinguished. It is also evident from TABLE 10, where the number of "clean" tweets incorrectly labeled as "hateful" exceeds that of "offensive" tweets.

CONCLUSION

We presented a novel approach to identify hate speech on Twitter in this study. To automatically classify tweets into hateful, offensive, and clean categories, our proposed method uses emotive and semantic elements in addition to the most prevalent unigrams and hate speech patterns. With our suggested method, we can accurately categorize tweets as either offensive or non-offensive (a binary classification) with an accuracy of 87.4 percent, and as hateful, offensive, or clean (a ternary classification) with an accuracy of 78.5 percent. Our long-term goal is to develop a more comprehensive hate speech pattern dictionary that, in conjunction with a unigram dictionary, can identify offensive and racist content on the internet. We will conduct a quantitative research to determine the prevalence of hate speech across various demographics, including gender, age, geography, etc.

REFERENCES

[1] R. D. King and G. M. Sutton, ``High times for hate crimes: Explaining the temporal clustering of hate-motivated offending," *Criminology*, vol. 51, no. 4, pp. 871_894, 2013.

[2] J. P. Breckheimer, ``A haven for hate: The foreign and domestic implica- tions of protecting Internet hate speech under the _rst amendment," *South California Law Rev.*, vol. 75, no. 6, p. 1493, Sep. 2002.

[3] P. Burnap and M. L. Williams, ``Cyber hate speech on twitter: An appli- cation of machine classi_cation and statistical modeling for policy and decision making," *Policy Internet*, vol. 7, no. 2, pp. 223_242, Jun. 2015.

[4] A. H. Razavi, D. Inkpen, S. Uritsky, and S. Matwin, ``Offensive lan- guage detection using multi-level classi_cation," *Advances in Arti_- cial Intelligence*, vol. 6085. Ottawa, ON, Canada: Springer, Jun. 2010, pp. 16_27.

[5] W. Warner and J. Hirschberg, ``Detecting hate speech on the world wide Web," in *Proc. 2nd Workshop Lang. Social Media*, Jun. 2012, pp. 19_26.

[6] M. Bouazizi and T. O. Ohtsuki, ``A pattern-based approach for sarcasm detection on Twitter," *IEEE Access*, vol. 4, pp. 5477_5488, 2016.

[7] D. Davidov, O. Tsur, and A. Rappoport, ``Semi-supervised recognition of sarcastic sentences in Twitter and Amazon," in *Proc. 14th Conf. Comput. Natural Lang. Learn.*, Jul. 2010, pp. 107_116.

[8] M. Bouazizi and T. Ohtsuki, ``Sentiment analysis: From binary to multi- class classi_cation: A pattern-based approach for multi-class sentiment analysis in Twitter," in *Proc. IEEE ICC*, May 2016, pp. 1_6.
[9] M. Bouazizi and T. Ohtsuki, ``Sentiment analysis in Twitter: From clas- si_cation to quanti_cation of sentiments within tweets," in *Proc. IEEE GLOBECOM*, Dec. 2016, pp. 1_6.
[10] J. M. Soler, F. Cuartero, and M. Roblizo, ``Twitter as a tool for pre- dicting elections results," in *Proc. IEEE/ACM ASONAM*, Aug. 2012, pp. 1194_1200.
[11] S. Homoceanu, M. Loster, C. Lo_, and W.-T. Balke, ``Will I like it? Providing product overviews based on opinion excerpts," in *Proc. IEEE CEC*, Sep. 2011, pp. 26_33.
[12] U. R. Hodeghatta, ``Sentiment analysis of Hollywood movies on Twitter," in *Proc. IEEE/ACM ASONAM*, Aug. 2013, pp. 1401_1404.
[13] Z. Zhao, P. Resnick, and Q. Mei, ``Enquiring minds: Early detection of rumors in social media from enquiry posts," in *Proc. Int. Conf.WorldWide Web*, May 2015, pp. 1395_1405.
[14] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, ``Hate speech detection with comment embeddings," in *Proc. WWW Companion*, May 2015, pp. 29_30.
[15] N. D. Gitari, Z. Zuping, H. Damien, and J. Long, ``A lexicon-based approach for hate speech detection," *Int. J. Multimedia Ubiquitous Eng.*, vol. 10, no. 4, pp. 215_230, Apr. 2015.
[16] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, ``Abusive language detection in online user content," in *Proc. WWW*, Apr. 2016, pp. 145_153.
[17] I. Kwok and Y.Wang, ``Locate the hate: Detecting tweets against blacks," in *Proc. AAAI*, Jul. 2013, pp. 1621_1622.
[18] Z. Waseem and D. Hovy, ``Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter," in *Proc. Student Res. Workshop (NAACL)*, Jun. 2016, pp. 88_93.
[19] T. Davidson, D. Warmsley, M. Macy, and I. Weber, ``Automated hate speech detection and the problem of offensive language," in *Proc. ICWSM*, May 2017, pp. 1_4.
[20] L. Derczynski, A. Ritter, S. Clark, and K. Bontcheva, ``Twitter part-of- speech tagging for all: Overcoming sparse and noisy data," in *Proc. Int. Conf. RANLP*, Sep. 2013, pp. 198_206.
[21] S. Das and M. Chen, ``Yahoo! for Amazon: Extracting market sentiment from stock message boards," in *Proc. 8th Asia Paci_c Finance Assoc. Annu. Conf.*, vol. 35. Jul. 2001, p. 43.
[22] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, ``The WEKA data mining software: An update," *ACM SIGKDD Explorations Newslett.*, vol. 11, no. 1, pp. 10_18, Jun. 2009.