Detecting Offensive Language in Social Media to Protect Adolescent Online Safety

Ying Chen[1], Sencun Zhu[1,3]
[1]Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA, USA
{yxc242, szhu}@cse.psu.edu

Yilu Zhou[2]
[2]Department of Information Systems and Technology Management
George Washington University
Washington, DC, USA
yzhou@gwu.edu

Heng Xu[3]
[3]College of Information Sciences and Technology
The Pennsylvania State University,
University Park, PA, USA
hxu@ist.psu.edu

***Abstract*—Since the textual contents on online social media are highly unstructured, informal, and often misspelled, existing research on message-level offensive language detection cannot accurately detect offensive content, and user-level offensiveness evaluation is still an under researched area. To bridge this gap, we propose the Lexical Syntactic Feature (LSF) architecture to detect offensive content and identify potential offensive users in social media. We distinguish the contribution of pejoratives/profanities and obscenities in determining offensive content, and introduce hand-authoring syntactic rules in identifying name-calling harassments. In particular, we incorporate a user's writing style, structure and specific cyberbullying content as features to predict the user's potentiality to send out offensive content. Results from experiments showed that our LSF framework performed significantly better than existing methods in offensive content detection. It achieves precision of 98.24% and recall of 94.34% in sentence offensive detection, as well as precision of 77.9% and recall of 77.8% in user offensive detection. Meanwhile, the processing speed of LSF is approximately 10msec per sentence, suggesting the potential for effective deployment in social media.**

Keywords – cyberbullying; adolescent safety; offensive languages; social media

I. INTRODUCTION

With the rapid growth of social media, users especially adolescents are spending significant amount of time on various social networking sites to connect with others, to share information, and to pursue common interests,. In 2011, 70% of teens use social media sites on daily basis [1] and nearly one in four teens hit their favorite social-media sites 10 or more times a day [2]. While adolescents benefit from their use of social media by interacting with and learning from others, they are also at the risk of being exposed to large amounts of offensive online contents. ScanSafe's monthly "Global Threat Report" [3] found that up to 80% of blogs contained offensive contents and 74% included porn in the format of image, video, or offensive languages. In addition, cyber-bullying occurs via offensive messages posted on social media. It has been found that 19% of teens report that someone has written or posted mean or embarrassing things about them on social networking sites [1]. As adolescents are more likely to be negatively affected by biased and harmful contents than adults, detecting online offensive contents to protect adolescent online safety becomes an urgent task. The Children's Internet Protection Act (CIPA) was enacted in early 2001 to address concerns on children's access to visual offensive content over Internet. While CIPA concerns about image contents, offensive languages in the form of unstructured and despicable texts can be as harmful as multimedia materials. To comply with CIPA requirements, administrators of social media often manually review online contents to detect and delete offensive materials. However, the manual review tasks of identifying offensive contents are labor intensive, time consuming, and thus not sustainable and scalable in reality. Some automatic content filtering software packages, such as Appen and Internet Security Suite, have been developed to detect and filter online offensive contents. Most of them simply blocked webpages and paragraphs that contained dirty words. These word-based approaches not only affect the readability and usability of web sites, but also fail to identify subtle offensive messages. For example, under these conventional approaches, the sentence "you are such a crying baby" will not be identified as offensive content, because none of its words is included in general offensive lexicons. In addition, the false positive rate of these word-based detection approaches is often high, due to the word ambiguity problem, i.e., the same word can have very different meanings in different contexts. Moreover, existing methods treat each message as an independent instance without tracing the source of offensive contents.

To address these limitations, we propose a more powerful solution to improve the deficiency of existing offensive content detection approaches. Specifically, we propose the Lexical Syntactic Feature-based (LSF) language model to effectively detect offensive language in the social media to protect adolescents. LSF provides high accuracy in subtle offensive message detection, and it can reduce the false positive rate. Besides, LSF not only examines messages, but also the person who posts the messages and his/her patterns of posting. LSF can be implemented as a client-side application for individuals and groups who are concerned about adolescents' online safety. It is able to detect whether online users and websites push recognizable offensive contents to adolescents, trigger applications to alert the senders to regulate their behavior, and eventually block the sender if this pattern continues. Users are also allowed to adjust the threshold of acceptable level of offensive contents. Our language model may not be able to make adolescents completely immune to offensive contents, because it is hard to fully detect what is "offensive." However, we aim to

provide an improved automatic tool to detect offensive contents in social media to help school teachers and parents have better control over the contents adolescents are viewing.

While there is no universal definition of "offensive," in this study we employ Jay and Janschewitz's [4] definition of offensive language as vulgar, pornographic, and hateful language. Vulgar language refers to coarse and rude expressions, which include explicit and offensive reference to sex or bodily functions. Pornographic language refers the portrayal of explicit sexual subject matter for the purposes of sexual arousal and erotic satisfaction. Hateful language includes any communication outside the law that disparages a person or a group on the basis of some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, and religion. All of these are generally immoral and harmful for adolescents' mental health.

II. RELATED WORK

In this section, we review existing methods on offensive content filtering in social media, and then focus on text mining based offensive detection research.

A. *Offensiveness Content Filtering Methods in Social Media*

Popular online social networking sites apply several mechanisms to screen offensive contents. For example, Youtube's safety mode, once activated, can hide all comments containing offensive languages from users. But pre-screened content will still appear—the pejoratives replaced by asterisks, if users simply click "Text Comments." And on Facebook, users can add comma-separated keywords to the "Moderation Blacklist." When people include blacklisted keywords in a post and/or a comment on a page, the content will be automatically identified as spam and thus be screened. Twitter client, "Tweetie 1.3," was rejected by Apple Company for allowing foul languages to appear in users' tweets. Currently, Twitter does not pre-screen users' posted contents, claiming that if users encounter offensive contents, they can simply block and unfollow those people who post offensive contents. In general, the majority of popular social media use simple lexicon-based approach to filter offensive contents. Their lexicons are either predefined (such as Youtube) or composed by the users themselves (such as Facebook). Furthermore, most sites rely on users to report offensive contents to take actions. Because of their use of simple lexicon-based automatic filtering approach to block the offensive words and sentences, these systems have low accuracy and may generate many false positive alerts. In addition, when these systems depend on users and administrators to detect and report offensive contents, they often fail to take actions in a timely fashion. For adolescents who often lack cognitive awareness of risks, these approaches are hardly effective to prevent them from being exposed to offensive contents. Therefore, parents need more sophisticate software and techniques to efficiently detect offensive contents to protect their adolescents from potential exposure to vulgar, pornographic and hateful languages.

B. *Using Text Mining Techniques to Detect Online Offensive Contents*

Offensive language identification in social media is a difficult task because the textual contents in such environment is often unstructured, informal, and even misspelled. While defensive methods adopted by current social media are not sufficient, researchers have studied intelligent ways to identify offensive contents using text mining approach. Implementing text mining techniques to analyze online data requires the following phases: 1) data acquisition and preprocess, 2) feature extraction, and 3) classification. The major challenges of using text mining to detect offensive contents lie on the feature selection phrase, which will be elaborated in the following sections.

a) Message-level Feature Extraction

Most offensive content detection research extracts two kinds of features: lexical and syntactic features.

Lexical features treat each word and phrase as an entity. Word patterns such as appearance of certain keywords and their frequencies are often used to represent the language model. Early research used Bag-of-Words (BoW) in offensiveness detection[5]. The BoW approach treats a text as an unordered collection of words and disregards the syntactic and semantic information. However, using BoW approach alone not only yields low accuracy in subtle offensive languagedetection, but also brings in a high false positive rate especially during heated arguments, defensive reactions to others' offensive posts, and even conversations between close friends. N-gram approach is considered as an improved approach in that it brings words' nearby context information into consideration to detect offensive contents[6]. N-grams represent subsequences of N continuous words in texts. Bi-gram and Tri-gram are the most popular N-grams used in text mining. However, N-gram suffers from difficulty in exploring related words separated by long-distances in texts. Simply increasing N can alleviate the problem but will slow down system processing speed and bring in more false positives.

Syntactic features: Although lexical features perform well in detecting offensive entities, without considering the syntactical structure of the whole sentence, they fail to distinguish sentences' offensiveness which contain same words but in different orders. Therefore, to consider syntactical features in sentences, natural language parsers[7] are introduced to parse sentences on grammatical structures before feature selection. Equipping with a parser can help avoid selecting un-related word sets as features in offensiveness detection.

b) User-level Offensiveness Detection

Most contemporary research on detecting online offensive languages only focus on sentence-level and message-level constructs. Since no detection technique is 100% accurate, if users keep connecting with the sources of offensive contents (e.g., online users or websites), they are at

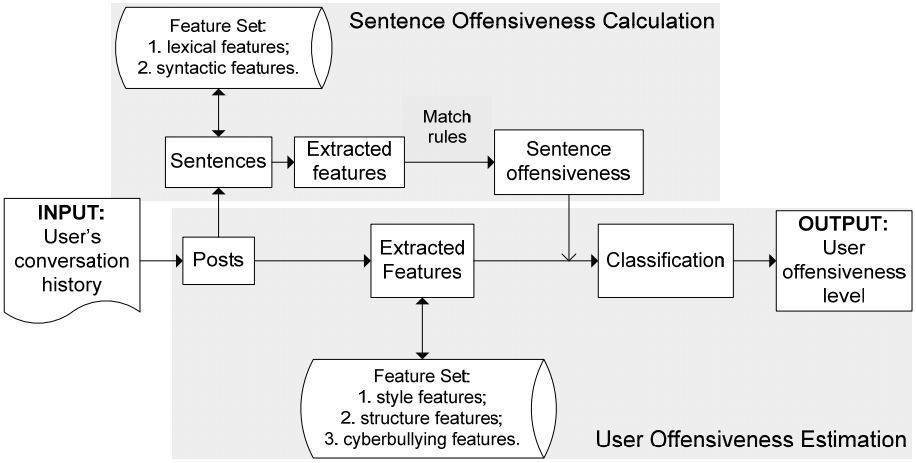

Figure 1. Framework of LSF-based offensive language detection

high risk of continuously exposure to offensive contents. However, user-level detection is a more challenging task and studies associated with the user level of analysis are largely missing. There are some limited efforts at the user level. For example, Kontostathis et al [8] propose a rule-based communication model to track and categorize online predators. Pendar [6] uses lexical features with machine learning classifiers to differentiate victims from predators in online chatting environment. Pazienza and Tudorache [9] propose utilizing user profiling features to detect aggressive discussions. They use users' online behavior histories (e.g., presence and conversations) to predict whether or not users' future posts will be offensive. Although their work points out an interesting direction to incorporate user information in detecting offensive contents, more advanced user information such as users' writing styles or posting trends or reputations has not been included to improve the detection rate.

III. Research Questions

Based on our review, we identify the following important research questions to prevent adolescents from offensive textual content:

- How to design an effective framework that incorporates message-level features and user-level features to detect and prevent offensive content in social media?
- What strategy is effective in detecting and evaluating level of offensiveness in a message? Will advanced linguistic analysis improve the accuracy and reduce false positives in detecting message-level offensiveness?
- What strategy is effective in detecting and predicting user-level offensiveness? Besides using information from message-level offensiveness, could user profile information further improve the performance?
- Is the proposed framework efficient enough to be deployed on real time social media?

IV. Design Framework

In order to tackle these challenges, we propose a Lexical Syntactic Feature (LSF) based framework to detect offensive content and identify offensive users in social media. We propose to include two phases of offensiveness detection. Phase 1 aims to detect the offensiveness on the sentence level and Phase 2 derives offensiveness on the user level. In Phase 1, we apply advanced text mining and natural language processing techniques to derive lexical and syntactic features of each sentence. Using these features, we derive an offensive value for each sentence. In Phase 2, we further incorporate user-level features where we leverage research on authorship analysis. The framework is illustrated in Fig.1.

The system consists of pre-processing and two major components: sentence offensiveness prediction and user offensiveness estimation. During the pre-processing stage, users' conversation history are chunked into posts, and then into sentences. During sentence offensiveness prediction, each sentence's offensiveness can be derived from two features: its words' offensiveness and the context. We use lexical feature to represent words offensiveness in a sentence, and syntactic feature to represent context in a sentence. Words' nature offensiveness is measured from two lexicons. For the context, we grammatically parse sentences into dependency sets to capture all dependency types between a word and other words in the same sentence, and mark some of its related words as intensifiers. The intensifiers are effective in detecting whether offensive words are used to describe users or other offensive words. During user offensiveness estimation stage, sentence offensiveness and users' language patterns are helped to predict users' likelihood of being offensive.

Sentence Offensiveness Calculation

To bridge the gap of the previous methods for sentence offensiveness detection [10-13], we propose a new method of sentence-level analysis based on offensive word lexicons and sentence syntactic structures. Firstly, we construct two offensive word dictionaries based on different strengths of offensiveness. Secondly, the concept of syntactic intensifier

is introduced to adjust words' offensiveness levels based on their context. Lastly, for each sentence, an offensiveness value is generated by aggregating its words' offensiveness. Since we already use intensifiers to further adjust words' offensiveness, no extra weights are assigned to words during the aggregation.

a) Lexical features: Offensiveness Dictionary Construction

Offensive sentences always contain pejoratives, profanities, or obscenities. Strongly profanities, such as "f***" and "s***", are always undoubtedly offensive when directed at users or objects; but there are many other weakly pejoratives and obscenities, such as "stupid" and "liar," that may also be offensive. This research differentiates between these two levels of offensiveness based on their strength. The offensive word lexicon used in this research includes the lexicon used in Xu and Zhu's study [14] and a lexicon, based on Urban Dictionary, established during the coding process. All profanities are labeled as strongly offensive. Pejoratives and obscenities receive the label of strongly offensive if more than 80% of their use in our dataset is offensive. The dataset is collected from Youtube command board (details will be described in the experiment section). Otherwise, known pejoratives and obscenities receive the label of weakly offensive word. Word offensiveness is defined as: for each offensive word, w, in sentence, s, its offensiveness

$$O_w = \begin{cases} a_1 & \text{if w is a strongly offensive word} \\ a_2 & \text{if w is a weakly offensive word} \end{cases} \quad (1)$$

where $1 \geq a_1 \geq a_2$, for the offensiveness of strongly offensive words is higher than weakly offensive words.

b) Syntactic features: Syntactic Intensifier Detection

Once pejoratives or obscenities are directed at online users, or semantically associated with another pejorative or obscenity, they become more offensive from users' perspectives. For example, "you stupid" and "f***ing stupid," are much more insulting than "This game is stupid." In addition, the dataset from Content Analysis for the Web2.0 Workshop [1] shows that most offensive sentences include not only offensive words but also user identifiers, i.e. second person pronouns, victim's screen names, and other terms referring to people. Table I lists some examples of this type of sentences.

When offensive words grammatically relate to user identifiers or other offensive words in sentences, the offensiveness level requires adjusting. This study uses a

TABLE I. LANGUAGE FEATURES OF OFFENSIVE SENTENCES

Language Features	Example
Second person pronoun (victim's screen name) + pejorative (i.e. JK, gay, wtf, emo, fag, loner, loser)	<You, gay>
Offensive adjective (i.e. stupid, foolish, sissy) + people referring terms (i.e. emo, bitch, whore, boy, girl)	<stupid, bitch> <sissy, boy>

[1] http://caw2.barcelonamedia.org/

nature language process parser, proposed by Stanford Natural Language Processing Group, to capture the grammatical dependencies within a sentence. The parsing results of sentences become combinations of a dependency-type and word-pair with the form "(governor, dependent)". For example, the typed dependency "appos (you, idiot)" in the sentence "You, by any means, an idiot." means that "idiot", the dependent, is an appositional modifier of the pronoun "you," the governor. The governor and dependent can be any syntactic elements of sentences. Some selected dependency types capture the possible grammatical relations between an offensive word and a user-identifier (or another offensive word) in a sentence. The study also proposes syntactical intensifier detection rules listed in Table II (A represents a user identifier, and B represents an offensive word).

The offensiveness levels of offensive words and other inappropriate words receive adjustment by multiplying their prior offensiveness levels by an intensifier[15]. In sentence, s, if all words syntactically related to an offensive word, w, are categorized as a dependency set, $D_{w,s} = \{d_1, \ldots, d_k\}$, for each $d_j (1 \leq j \leq k)$ is defined as:

$$d_j = \begin{cases} b_1 & \text{if } d_j \text{ is a user identifier} \\ b_2 & \text{if } d_j \text{ is an offensive word} \end{cases} \quad (2)$$

where $b_1 \geq b_2 \geq 1$, for offensive words used to describe users are more offensive than the words used to describe other offensive words. Thus, the value of intensifier, I_w, for the offensive word, w, can be calculated as $\sum_1^k d$.

c) Sentence Level Offensiveness Value Generation

Consequently, the offensiveness value of sentence, s, becomes a determined linear combination of words' offensiveness, $O_s = \sum o_w I_w$.

User offensiveness estimation

In user offensiveness estimation stage, our design has two major steps: aggregating users' sentence offensiveness and extracting extra features from users' language styles. We incorporate sentence offensiveness values and user language features to classify users' offensiveness.

a) Sentence Offensiveness Aggregation

While there are few studies on user-level offensiveness analysis, studies on document-level sentiment analysis share some similarity with this research [15-18]. Document-level sentiment analysis predicts the overall polarity of a document by aggregating polarity scores of individual sentences. Since the importance of each sentence varies in a document, one assigns weights to all sentences to adjust their contributions to the overall polarity. Similarly, we cannot simply sum up the offensive values of all sentences to compute users' offensiveness, because the strength of sentence offensiveness

TABLE II. SYNTACTICAL INTENSIFIER DETECTION RULES

Rules	Meanings	Examples	Dependency Types
Descriptive Modifiers and complements: A(noun, verb, adj) ←B(adj, adv, noun)	B is used to define or modify A.	you f***ing; you who f***ing; you…the one…f***ing.	• abbrev (abbreviation modifier), • acomp (adjectival complement), • amod (adjectival modifier), • appos (appositional modifier), • nn (noun compound modifier), • partmod (participial modifier)
Object: B(noun, verb) ←A(noun)	A is B's direct or indirect object.	F*** yourselves; shut the f** up; f*** you idiot; you are an idiot; you say that f***...	• dobj (direct object), • iobj (indirect object), • nsubj (nominal subject)
Subject: A(noun)→B(noun, verb)	A is B's subject or passive subject.	you f***…; you are **ed… …f***ed by you…	• nsubj (nominal subject), • nsubjpass (passive nominal subject), • xsubj (controlling subject), • agent (passive verb's subject).
Close phrase, coordinating conjunction: A and B; …A, B…; …B, B…	A and B or two Bs are close to each other in a sentence, but be separated by comma or semicolon.	F** and stupid; you, idiot.	• conj (conjunct), • parataxis (from Greek for "place side by side")
Possession modifiers: A(noun)→B(noun)	A is a possessive determiner of B.	your f*** …; s*** falls out of your mouth.	• poss (holds between the user and its possessive determiner)
Rhetorical questions: A(noun)←B(noun)	B is used to describe clause with A as root (main object).	Do you have a point, f***?	• rcmod (relative clause modifier)

depends on its context. For example, one may post "Stupid guys need more care. You are one of them." If we calculate offensiveness level of this sentence without considering the context, the offensiveness of this post will not be detected even using natural language parsers. To bypass the limitation of current parsers, we modify each post by combining sentences and replacing the periods with commas before feeding them to parsers. Then the parser generates different phrase sets for further calculation of the offensiveness level of the modified posts. However, since the modified posts may sometimes miss the original meanings, we have to balance between using the sum of sentence offensiveness and using the offensiveness of the modified posts to represent post offensiveness. In this case, the greater value of the two is chosen to represent the final posts' offensiveness levels. The detail of the schema is illustrated as following:

Given a user, u, we retrieve his/her conversation history which contains m posts $\{p_1,...,p_m\}$, and each post $p_i(1 \le i \le m)$ contains sentences $\{s_1,...,s_n\}$. Sentence offensiveness values are denoted as $\{O_{s_1},...,O_{s_n}\}$. The original offensiveness value of post p, $O_p = \sum O_s$. The offensiveness value of modified posts can be presented as, $O_{p\to s}$. So the final post offensiveness $O_p^{'}$ of post p can be calculated as, $O_p^{'} = \max(O_p, O_{p\to s}) = \max(\sum O_s, O_{p\to s})$. Hence, the offensiveness value, O_u, of user, u, can be presented as, $O_u = \frac{1}{m}\sum O_p^{'}$. We normalize the offensiveness value because users who have more posts are not necessarily more offensive than others. O_u, should be no less than 0.

b) Additional Features Extracted from Users' Lanuage Profiles

Other characteristics such as the punctuation used, sentence structure, and the organization of sentences within posts could also affect others' perceptions of the poster's offensiveness level. Considering the following cases:

Sentence styles. Users may use punctuation and words with all uppercase letters to indicate feelings or speaking volume. Punctuation, such as exclamation marks, can emphasize offensiveness of posts. (i.e. Both "You are stupid!" and "You are STUPID." are stronger than "You are stupid."). Some users tend to post short insulting comments, such as "Holy s***." and "You idiot." Consequently, compared to those who post the same number of offensive words but in longer sentences, the former users appear more offensive for intensive usage of pejoratives and obscenities. Users may use offensive words to defend themselves when they are arguing with others who are offensive. But it is costly to detect whether their conversation partners are offensive or not. Instead, we noticed that arguments should happen in relatively short period of time. For example, for user u, whose conversation history is valid in 100 days within 2 years, while the time period he/she is using offensive words is only 5 days, no matter how many offensive words (s)he is using, (s)he should not be considered as an offensive user. Thus, to make sure users' offensiveness values evenly distributed over the span of their conversation history is a reasonable way to differentiate general offensive users from the occasional ones.

TABLE III. ADDITIONAL FEATURE SELECTION FOR USER OFFENSIVENESS ANALYSIS

Style Features	Structural Features	Content-specific Features
-Ratio of short sentences -Appearance of punctuations -Appearance of words with all uppercase letters	-Ratio of imperative sentences -Appearance of offensive words as nouns, verbs, adjs and advs.	-Race -Religion -Violence -Sexual orientation -Clothes -Accent -Appearance -Intelligence -Special needs or disabilities

Sentence structures. Users who frequently use imperative sentences tend to be more insulting, because imperative sentences deliver stronger sentiments. For example, a user who always posts messages such as "F***ing u" and "Slap your face" gives the impression of being more offensive and aggressive than those ones posting "you are f***ing" and "your face get slapped."

Cyberbullying related content. O'Neill and Zinga [19] described seven types of children who, due to differences from peers, may be easy targets for online bullies, including those children from minority races, with religious beliefs, or with non-typical sexual orientations. Detecting online conversations referring to these individual differences also provides clues for identifying offensive users.

Based on the above observations, three types of features are developed to identify the level of offensiveness, which leveraged from authorship analysis research on cybercrime investigation [20-25]: style features, structural features, and content-specific features. Style features and structural features capture users' language patterns, while content-specific features help to identify abnormal contents in users' conversations. The style features in our study infer users' offensiveness levels from their language patterns, including whether or not they are frequently/recently using offensive words and intensifiers such as uppercase letters and punctuation. The structural features capture the way users construct their posts, which check whether or not users are frequently using imperative sentences. They also try to infer users' writing styles by checking offensive words used as nouns, verbs, adjs, or advs. The content-specific features check whether or not users post suspicious contents which probably will be identified as cyberbullying messages. In this study, we identify cyberbullying contents by checking whether they contain cyberbullying related words (i.e. religious words). The details of these features are summarized in Table III.

c) Overall User Offensiveness Estimation

Besides style features, structure features and content-specific features, sentence offensiveness values are considered as one type of user language features. By using these features, machine learning techniques can be adopted to classify users' offensiveness levels.

V. EXPERIMENT

This section describes several experiments we conducted to examine LSF on detecting offensiveness languages in social media.

Dataset Description

The experimental dataset, retrieved from Youtube comment boards, is a selection of text comments from postings in reaction to the top 18 videos. Classification of the videos includes thirteen categories: Music, Autos, Comedies, Educations, Entertainments, Films, Gaming, Style, News, Nonprofits, Animals, Sciences, and Sports. Each text comment includes a user id, a timestamp and text content. The user id identifies the author who posted the comment, the timestamp records when the comment was posted and the text content contained a user's comments. The dataset includes comments from 2,175,474 distinct users.

Pre-processing

Before feeding the dataset to the classifier, an automatic pre-processing procedure assembles the comments for each user and chunks them into sentences. For each sentence in the sample dataset, an automatic spelling and grammar correction process precedes introduction of the sample dataset to the classifier. With the help of WordNet corpus and spell-correction algorithm[2], correction of spelling and grammar mistakes in the raw sentences occurs by tasks such as deleting repeated letters in words, deleting meaningless symbols, splitting long words, transposing substituted letters, and replacing the incorrect and missing letters in words. As a result, words missing letters, such as "speling," are corrected to "spelling"; misspelled words, such as "korrect," change to "correct."

Experiment Settings in Sentence Offensive Prediction

The experiment compares six approaches in sentence offensive prediction:

a) Bag-of-words (BoW): The BoW approach disregards grammar and word order and detects offensive sentences by checking whether or not they contain both user identifiers and offensive words. This approach also acts as a benchmark.

b) 2-gram: The N-gram approach detects offensive sentences by selecting all sequences of n words in a given sentence and checking whether or not the sequences include both user identifiers and offensive words. In this approach, N equals to 2, it also acts as a benchmark.

c) 3-gram: N-gram approach, selecting all sequences of 3 words in a given sentence. It also acts as a benchmark.

d) 5-gram: N-gram approach, selecting all sequences of 5 words in a given sentence. It also acts as a benchmark.

e) Appraisal approach: The appraisal approach was proposed for sentiment analysis[26], and here we use it on sentence offenisve detection for comparision. It can detect offensive sentences by going through all types of dependency sets and checking whether or not certain offensive words and user identifiers are grammatically related in a given sentence. The major differences between applying the appraisal approach on sentence offensive detection and ours is that appraisal approach cannot differenciate offensive words based on their strength, and it generally considers two words as "related" if they are within

[2] Spell-Correction Algorithm, at http://norvig.com/spell-correct.html

any type dependency set, while some of the dependency type does not really indicate one is acting on the other. For instance, type dependency "parataxis" relation (from Greek for "place side by side") is a relation between the main verb of a clause and other sentential elements, such as a sentential parenthetical, a clause after a ":" or a ";". An example sentence for type dependency "parataxis(left, said)" can be "The guy, John said, left early in the morning". Here "said" and "left" are not really used to describe one another.

f) LSF: The sentence offensive prediction method proposed in this study.

Evaluation Metrics

In our experiments, standard evaluation metrics for classification in sentiment analysis [16, 17, 27] (i.e., precision, recall, and f-score) are used to evaluate the performance of LSF. In particular, precision presents the percent of identified posts that are truly offensive messages. Recall measures the overall classification correctness, which represents the percent of actual offensive messages posts that are correctly identified. False positive (FP) rate represents the percent of identified posts that are not truly offensive messages. False negative (FN) rate represents the percent of actual offensive messages posts that are unidentified. F-score [13] represents the weighted harmonic mean of precision and recall, which is defined as:

$$f-score = \frac{2(precision \times recall)}{precision + recall} \quad (3)$$

Experiment 1: Sentence Offensiveness Calculation

In this experiment, we randomly select a uniform distributed sample from the complete dataset, which includes 1700 sentences. In total, we select 359 strongly offensive words and 251 weakly offensive words as offensive word lexicons, and the experimental parameters are set as: $a_1 = 1; a_2 = 0.5; b_1 = 2; b_2 = 1.5$. We define "1" to be the threshold for offensive sentence classification; that is, sentences with offensiveness values more than (inclusive) "1" receive labels of offensive sentences, because by our definition, offensive sentence means a sentence containing strongly offensive words, or containing weakly offensive words used to describe another user. After manual labeling, 173 sentences are marked as "offensive". Subsequently, a manual check on the classifier's output produced the results as shown in Fig. 2.

According to Fig.2, none of the baseline approaches provides recall rate higher than 70%, because many of the offensive sentences are imperatives, which omit all user identifiers. Among the baseline approaches, the BoW approach has the highest recall rate 66%. However, BoW generates a high false positive rate because it captures numbers of unrelated <user identifier, offensive word> sets. The recall of N-gram is low when n is small. However, as n increases, the false positive rate increases as well. Once N equals to the length of sentences, N-gram is equivalent to the bag-of-words approach. To further apply N-gram in the classification, application of different values of N is necessary to balance, perfectly, the trade-off between recall and false positive rate.

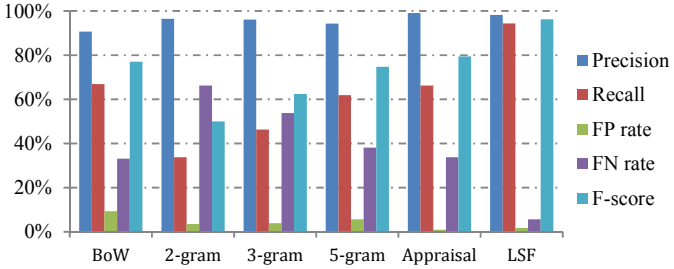

Figure 2. Accuracies of sentence level offensiveness detection

The appraisal approach reaches high precision, but its recall rate is poor. LSF obtains the highest f-score, because it sufficiently balances the precision-recall tradeoff. It achieves precision of 98.24% and recall of 94.34% in sentence offensive detection. Unfortunately, the parser sometimes misidentifies noun appositions, in part because of typographical errors in the input, such as: "you stupid sympathies" Here, the sender presumably meant to write "your" instead of "you." This is the major reason for false negative rates. The false positive rate arises mainly from multiple appearances of weak offensive words, for example, "fake and stupid," which can only represent a negative opinion for a video clip but accidently identified as "offensive" because LSF calculate a value higher than (or equal to) 1.

Experiment 2: User Offensiveness Estimation-with presence of strongly offensive words

In this experiment we randomly selected 249 users with uniformly distributed offensiveness values calculated from Experiment 1 from the dataset. The selected users have 15 posts on average. Each of the 249 users was rated by three volunteers from computer science and information science department who were not otherwise involved in this research. Each user was rated by two males and one female. Volunteers were told to mark a user as being offensive if his(her) posts contained insulting or abusive language which makes recipient feel offended, not merely if the sender expressed disagreement with the recipient. Volunteers could classify a message with "offensive" or "inoffensive". The cronbach's α value (agreement rate) of the volunteers is 0.73. A valid user label was generated when all of the volunteers put the same label on that user. After balancing the positive and negative results, we have 99 users in each class.

Machine learning techniques—NaiveBayes (NB) and SVM—are used to perform the classification, and 10-fold cross validation was conducted in this experiment. To fully evaluate the effectiveness of users' sentence offensiveness value (LSF), style features, structure features and content-specific features for user offensiveness estimation, we fed them sequentially into the classifiers, and get the result in Fig.3. The "Strong+Weak" means simply using offensive words as the base feature to detect offensive users. Similarly, "LSF" means the sentence offensiveness value generated by LSF is used as the basic feature.

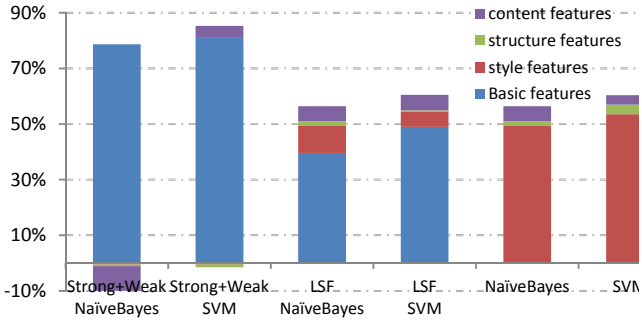

Figure 3. F-score for different feature sets using NB and SVM

According to Fig.3, offensive words and user language features are not compensating to each other to improve the detection rate, which means they are not independent. In contrast, incorporating with user language features, the classifiers have better detection rate than just adopting LSF. While all three types of features are useful to improve the classification rate, style features and content features are more valuable than structure features in user offensiveness classification. However, LSF is not as useful as using offensive words alone in detecting offensive user. One possible reason is that once the number of strongly offensive words beyond certain amount, the user who posts the comments is considered being offensive anyway. In such case, LSF might be less useful than using merely offensive words. We looked further into this situation and test the model under a situation where the messages does not contain strong offensive words and are not obviously offensive in Experiment 3.

Experiment 3: User Offensiveness Estimation-without strongly offensive words

In this experiment we only want to test the situation when the offensiveness of a user is subtle. We chose to use a dataset without strongly offensive words. Our testing data are randomized selections of the original data followed by filtering out messages that contain strong offensive words. We got 200 users with uniformly distributed offensiveness values. This dataset does not overlap with the one in experiment 2. The selected users have 85 posts on average, and none of the posts contains strongly offensive words. After balancing the positive and negative results, we have 81 users in each class. The experiment condition is identical to Experiment 2. The result in presented in Fig.4. "Weak" means it is simply using (weak) offensive words as the base feature to detect offensive users, because there is no strongly offensive words in this experiment.

According to Fig.4, offensive words and user language features are still not well compensating to each other to improve the detection rate, either as LSF and user language features. Hence, the dependency between user language features and offensive words, and the dependency between user language features and LSF are both data driven; they vary from domain to domain. Style features and content features are still more valuable than structure features in user offensiveness classification. However, we did observe the appearance of imperative sentences frequently occurs in offensive users' conversations. One reason to cause this is

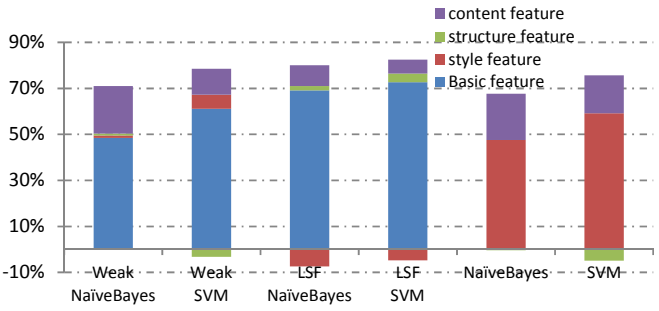

Figure 4. F-score for different feature sets using NB and SVM (without strongly offensive words)

the POS tagger does not have enough accuracy in tagging verbs, and it even marks "Yes", "Youre" and "Im" as verbs in some sentences. In such case, many imperative sentences are not be tagged, and the tagged ones are not necessary imperative.

In this experiment, LSF performs better than offensive words in detecting offensive user this time, it achieves precision of 77.9% and recall of 77.8% in user offensive detection using SVM, which proves our hypotheses that LSF do work better than using offensive words alone in detecting non-obvious user offensiveness detection, and incorporating user language features will further stir the detection rate. Therefore, we can further conclude that considering context and talking objects will help precisely detect offensive language which does not have dirty words. However, strong offensive word is still the primary element which annoys general readers. Our experiment results might suggest a possible 2-stage offensiveness detection when there are many appearance of strong offensive words.

Experiment 4: Efficiency

Experiment 4(a): Efficiency of Sentence Offensiveness Calculation

In addition to accuracy measurement, assessment of processing speed on masses of text messages is necessary, because speed is a critical attribute for offensive detection in real-time online communities. The sentence processing time in each case appear in Fig.5.

The average time for reading each word is 0.0002 ms, and it takes 0.0033 ms to compare it with the words in dictionaries to determine whether it is a user identifier or an offensive word. In our sample, each sentence contains about 10.42 words. Thus, the average processing time for BoW and N gram can be calculated as read time plus twice comparison time for each word in the sentence, which is about 0.07 ms

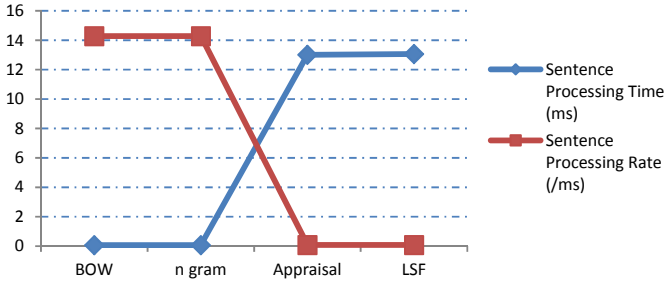

Figure 5. Sentence Processing Time for different methods

(shown in Fig.5). However, for appraisal approach, it takes longer time to grammatically parse sentences before the analysis. In contrast, LSF method firstly check whether sentence contain offensive words. If it does contain offensive words, LSF will proceed to parse the sentence and search for their intensifiers. We list the worst case for LSF method in Fig.5, and its performance really depend on the offensive sentence ratio on social media. However, we still can prove it is practical for application to online social media and other real-time online communities. Take Youtube as example, over 80% of its content don't contain offensive words, so the sentence processing rate for LSF can be cut down to 2.6 ms.

Experiment 4(b): Efficiency of User Offensiveness Estimation

In experiment 2, users have 15 posts on average, and each post contains 2 sentences, total 31 sentences posted by each user. In experiment 3, users have 85 posts on average, and each post contains 4 sentences, total 339 sentences posted by each user. The feature extraction time for different feature sets in experiment 2 and experiment 3 are presented in Fig.6.

From Fig.6, we find that aggregating users' sentences offensiveness (LSF) takes most of the time, and it is positive correlated with the number of sentences a user posts. Other than that, the calculation of structure features also takes much more time than style features and content-specific features. Assume an online user has 100 sentences in his (her) conversation history; it takes approximately 1.9s to extract both the sentence feature and language features, which will not even be noticed.

We further examined the classification rates for different feature sets using NaiveBayes and SVM classifiers. Since the rates vary from time to time, we run each instance 5 times

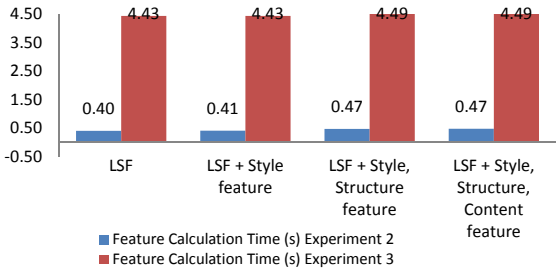

Figure 6. Feature extraction time for different feature sets in Experiment 2 and Experiment 3 (per user)

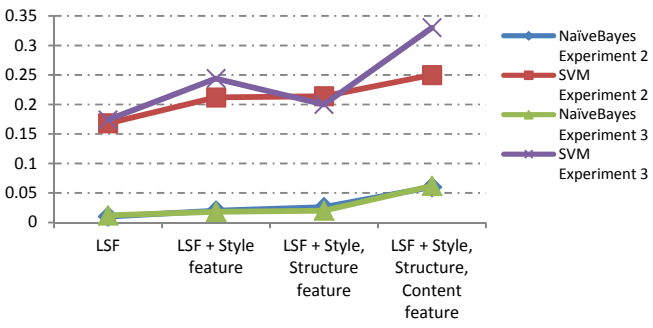

Figure 7. Classification Time for different feature sets using NaiveBayes and SVM classifiers in Experiment 2 and Experiment 3

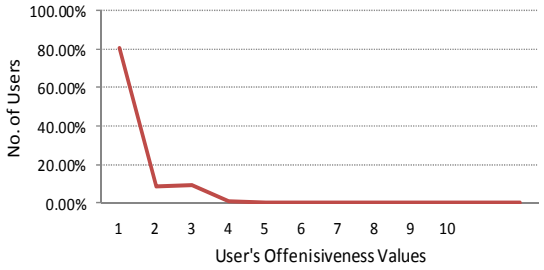

Figure 8. Users' offensiveness distribution on Youtube

and take the average. The result is shown in Fig.7. As to Fig.7, we find that the calculation rate of machine learning techniques is much faster than feature extraction time in Fig.6, the longest running time for machine learning classifiers is only 0.33s to predict users' offensiveness. And the classification rate is independent on the number of users and the number of sentences. Generally, NaiveBayes works much faster than SVM in classification, but SVM produces more accurate classification results.

To sum up, for a user who posts 100 sentences on social media, LSF takes approximately 2.2 second to predict users' offensive potential.

Experiment 5: Distribution of Offensive Content on Youtube

When we check users' offensiveness distributions on Youtube (Fig.8), we find that 81% of Youtube users do not use any offensive words in their posts, and the portion of users with offensiveness value over four is less than 1%. It indicates users' offensiveness values on the Youtube website satisfy the power law, and the number of users and their offensiveness values are negatively correlated.

VI. CONCLUSION

In this study, we investigate existing text-mining methods in detecting offensive contents for protecting adolescent online safety. Specifically, we propose the Lexical Syntactical Feature (LSF) approach to identify offensive contents in social media, and further predict a user's potentiality to send out offensive contents. Our research has several contributions. First, we practically conceptualize the notion of online offensive contents, and further distinguish the contribution of pejoratives/ profanities and obscenities in determining offensive contents, and introduce hand-authoring syntactic rules in identifying name-calling harassment. Second, we improved the traditional machine learning methods by not only using lexical features to detect offensive languages, but also incorporating style features, structure features and context-specific features to better predict a user's potentiality to send out offensive content in social media. Experimental result shows that the LSF sentence offensiveness prediction and user offensiveness estimate algorithms outperform traditional learning-based approaches in terms of precision, recall and f-score. It also achieves high processing speed for effective deployment in social media. Besides, the LSF tolerates informal and misspelling contents, and it can easily adapt to any formats of English writing styles. We believe that such language processing model will greatly help online offensive language monitoring, and eventually build a safer online environment.

ACKNOWLEDGMENT

We thank the reviewers for the valuable comments. This work of Ying Chen and Sencun Zhu was supported by NSF CAREER 0643906. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of NSF or the U.S. Government.

REFERENCES

[1] T. Johnson, R. Shapiro, and R. Tourangeau. (2011, 11/7). National survey of American attitudes on substance abuse XVI: Teens and parents. Available: http://www.casacolumbia.org/templates/NewsRoom.aspx?articleid=648&zoneid=51

[2] S. O. K. Gwenn, C.-P. Kathleen, and C. O. C. A. MEDIA, "Clinical report--the impact of social media on children, adolescents, and families.," Pediatrics, 2011.

[3] J. Cheng. (2007, 11/7). Report: 80 percent of blogs contain "offensive" content. Available: http://arstechnica.com/security/news/2007/04/report-80-percent-of-blogs-contain-offensive-content.ars

[4] T. Jay and K. Janschewitz, "The pragmatics of swearing," Journal of Politeness Research. Language, Behaviour, Culture, vol. 4, pp. 267-288, 2008.

[5] A. McEnery, J. Baker, and A. Hardie, "Swearing and abuse in modern British English," presented at the Practical Applications of Language Corpora, Peter Lang, Hamburg, 2000.

[6] N. Pendar, "Toward spotting the pedophile telling victim from predator in text chats," in Proceedings of the First IEEE International Conference on Semantic Computing, 2007, pp. 235-241.

[7] M.-C. d. Marneffe, B. MacCartney, and C. D. Manning, "Generating typed dependency parses from phrase structure parses," presented at the LREC, 2006.

[8] A. Kontostathis and L. E. A. Leatherman, "Chatcoder: Toward the tracking and categorization of internet predators," In Proc. Text Mining Workshop 2009 held in conjunction with the Ninth SIAM International Conference on Data Mining, 2009.

[9] M. Pazienza and A. Tudorache, "Interdisciplinary contributions to flame modeling," AI* IA 2011: Artificial Intelligence Around Man and Beyond, pp. 213-224, 2011.

[10] E. Spertus, "Smokey: Automatic recognition of hostile messages," Innovative Applications of Artificial Intelligence (IAAI) '97, 1997.

[11] A. Razavi, D. Inkpen, S. Uritsky, and S. Matwin, "Offensive language detection using multi-level classification," Advances in Artificial Intelligence, vol. 6085/2010, pp. 16-27, 2010.

[12] A. Mahmud, Ahmed, Kazi Zubair, and Khan, Mumit "Detecting flames and insults in text," in Proc. of 6th International Conference on Natural Language Processing (ICON' 08), 2008.

[13] D. Yin, Z. Xue, L. Hong, and B. Davison, "Detection of harassment on Web 2.0," in the Content Analysis in the Web 2.0 Workshop, 2009.

[14] Z. Xu and S. Zhu, "Filtering offensive language in online communities using grammatical relations," in Proceedings of The Seventh Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS'10), 2010.

[15] C. Zhang, D. Zeng, J. Li, F. Y. Wang, and W. Zuo, "Sentiment analysis of Chinese documents: from sentence to document level," Journal of the American Society for Information Science and Technology, vol. 60, pp. 2474-2487, 2009.

[16] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," In EMNLP'02: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, pp. 79-86, 2002.

[17] P. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews," In Proceedings of the Association for Computational Linguistics (ACL), pp. 417-424, 2002.

[18] B. K. Y. Tsou, R. W. M. Yuen, O. Y. Kwong, T. B. Y. Lai, and W. L. Wong, "Polarity classification of celebrity coverage in the Chinese press," Paper presented at the International Conference on Intelligence Analysis, 2005.

[19] T. O'Neill and D. Zinga, Children's rights: multidisciplinary approaches to participation and protection: Univ of Toronto Pr, 2008.

[20] R. Zheng, J. Li, H. Chen, and Z. Huang, "A framework for authorship identification of online messages: Writing-style features and classification techniques," Journal of the American Society of Information Science and Technology, vol. 57, pp. 378-393, 2006.

[21] J. V. Hansen, P. B. Lowry, R. D. Meservy, and D. M. McDonald, "Genetic programming for prevention of cyberterrorism through dynamic and evolving intrusion detection," Decision Support Systems, vol. 43, pp. 1362-1374, 2007.

[22] R. Zheng, Y. Qin, Z. Huang, and H. Chen, "Authorship analysis in cybercrime investigation," Intelligence and Security Informatics, pp. 959-959, 2010.

[23] S. Symonenko, E. D. Liddy, O. Yilmazel, R. Del Zoppo, E. Brown, and M. Downey, "Semantic analysis for monitoring insider threats," Intelligence and Security Informatics, pp. 492-500, 2004.

[24] A. Orebaugh and D. J. Allnutt, "Data mining instant messaging communications to perform author identification for cybercrime investigations," Digital Forensics and Cyber Crime, pp. 99-110, 2010.

[25] J. Ma, G. Teng, S. Chang, X. Zhang, and K. Xiao, "Social network analysis based on authorship identification for cybercrime investigation," Intelligence and Security Informatics, pp. 27-35, 2011.

[26] C. Whitelaw, N. Garg, and S. Argamon, "Using appraisal groups for sentiment analysis," in Proceedings of the 14th ACM International Conference on Information and Knowledge Management NY, USA, 2005, pp. 625-631.

[27] Q. Ye, W. Shi, and Y. Li, "Sentiment classification for movie reviews in Chinese by improved semantic oriented approach," in HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on System Sciences, 2006, pp. 53b-53b.