**Sentiment Analysis for Detecting Offensive**

**Expressions**

Casielyn Alivio, Joshua Dapitan, Anjanette Lasala, Casandra Miraballes, RiaSagum

College of Computer and Information Sciences

Polytechnic University of the Philippines

***Abstract ―*In this study, the researchers developed a system that will analyse sentences to determine if these sentences are offensive or not. Our approach was divided into two parts: pre-processing and sentiment analysis. We used Stanford NLP tools for some parts of the pre-processing. Our database for words is WordNet 2.1. We used randomly chosen comments from various social networking websites as input for the testing and experiment of the system. After the experiment is executed, considering 50 comments, the following rates was computed based from the gathered data. Recall – 60%, Precision – 66.67%, Specificity – 92.50% and Accuracy – 86%. We were able to conclude that using the definitions as the basis of sentiment analysis is possible to give possible results.**

***Key terms – Natural Language Processing, Sentiment Analysis, offensive***

I. INTRODUCTION

Online bullying is one of the most undesirable things that are in the internet. This is different from spam (that serves as distraction for communication). Online harassment is mostly driven by human moderation. Human moderation is costly. Other online harassment detectors merely analyse the lexemes. With this proposed system, this will assist moderators from moderating, and assisting internet based systems (such as chat and forums) to have a system that is user friendly and kid friendly.

The objective of this study is to develop a system that is able to determine whether a sentence is offensive or not by finding and determining the use of offensive words present in the sentence. The sentence is considered offensive if the offensive word or bad word is referred to a human like Cinderella is a faggot.

II. REVIEW OF RELATED WORKS

The researchers introduced the study of bullying to the NLP Community. Bullying, in both physical and cyber worlds (the latter known as cyberbullying), has been recognized as a serious national health issue among adolescents. One is being bullied or victimized when he or she is exposed repeatedly over time to negative actions on the part of others. There are wide ranges of emotions expressed in bullying traces. After manually inspecting a number of bullying traces in Twitter, our domain experts identified seven most common emotions such as anger, embarrassment, empathy, fear, pride, relief and sadness.

Analyzing Social Media to Detect Cyber Bullying using Sentiment Mining found that “sentiment analysis is the task of finding the opinions of people about specific textual entities. The decision making process of people is usually affected by the opinions formed by domain authorities and the proliferation of online discussions [1].

Sentiment Analysis has the potential to identify victims who pose high risk to themselves or others, and to enhance the scientific understanding of bullying overall Victims usually experience negative emotions such as depression, anxiety and loneliness. In extreme cases such emotions are more violent or even suicidal. Detecting at risk individuals via sentiment analysis enables potential interventions. In addition, social scientists are interested in sentiment analysis on bullying traces to understand participants’ motivations [2].

The Lexical Syntactical Feature (LSF) approach from the research [3] is to identify offensive contents in social media, and further predict a user’s potentiality to send out offensive contents. It includes two phases of offensive detection. Phase 1 aims to detect the offensiveness on the sentence level and Phase 2 derives offensiveness on the user level. In Phase 1, the researchers apply advanced text mining and natural language processing technique 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 system consists of pre-processing and two major components: sentence offensiveness prediction and user offensiveness estimation. During the pre-processing stage, user’s conversation history is chunked into posts, and then into sentences. During sentence offensiveness prediction, each sentence’s offensiveness can be derived from two features: its word’s offensiveness and the context. The researchers use lexical feature to represent words’ offensiveness in a sentence, and syntactic feature to represent context in a sentence. Words’ offensiveness nature 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 user’s likelihood of being offensive. 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.

Very few other research teams are working on the detection of cyber bullying. A misbehavior detection task was offered by the organizers of CAW 2.0, but only one submission was received. It is determined that a baseline text mining system (using bag of words approach) was significantly improved by including sentiment and contextual features. Even with the combined model, a support vector machine learner could only produce a recall level of 61.9% [4].

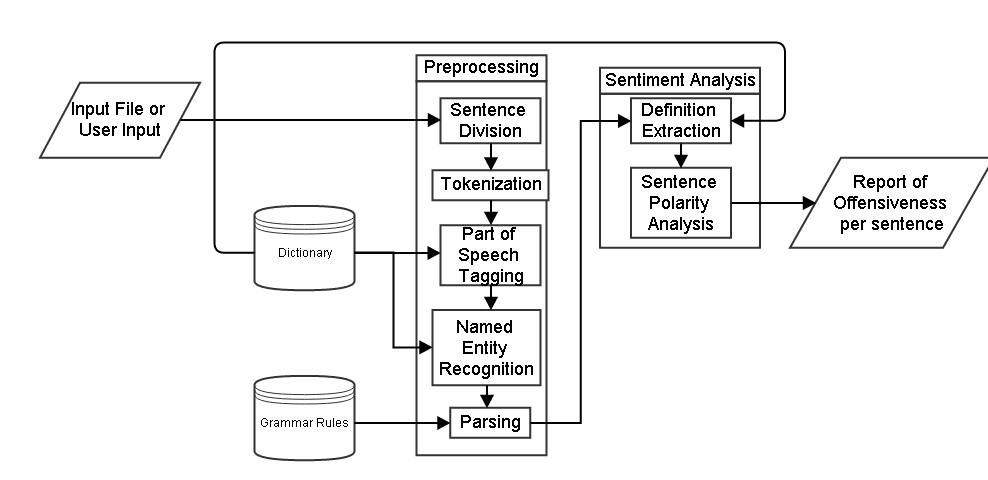
In the research [6], mainly tackles the problem about how the offensive language can be removed in a user message. They analyze the offensive language in text messages posted in online communities, and propose a new automatic sentence-level filtering approach that is able to semantically remove the offensive language utilizing the grammatical relations among words. Their solution includes 3 steps. First, they analyzed the characteristics of offensive text content in user messages. Then, they introduced their filtering philosophy according to the summarized characteristics. Finally, they show how this philosophy is transformed into heuristic rules applicable in the filtering process. The overview idea of their filtering approach is shown in the presented Algorithm 1 in which the functions POS tagging ad TD generator generate Part-of-Speech tags and typed dependency relations, respectively. They also use existing NLP (Natural Language Processing) tools to implement these two functions. They also focused in the design of two other functions CreateRelTree and EstimateRelTree. In their research assume that the filtering is based on a comprehensive offensive lexicon containing all offensive words. Words do not appear in the lexicon are considered inoffensive. Experiments their dataset, comments from Youtube, show over 90% agreement in filtered results between the proposed approach and manual filtering approach[6].

The researchers of [7] in proposed a novel semi-supervised approach for detecting profanity-related offensive content in Twitter. They introduced an approach that exploits linguistic regularities in profane language via statistical topic modeling on a huge Twitter corpus, and detects offensive tweets using these automatically generated features. Their step by step processes are as follows: (a) Bootstrap between twitters and tweets based on a seed word set to obtain training tweets for topic model learning; (b) Topic models are learned via a generative LDA approach; (c) Tweets in a holdout testing set are processed in the same fashion as in (a); (d) Topic distributions are inferred for each testing tweet by the topic model learned in step (b); (e) Seed words are applied against each testing tweet, leading to a binary lexicon feature; (f) ML models are built and evaluated. The keyword matching technique has been shown to perform very well in the literature and achieved a TP of 69.7% with an FP of 3.77% in our experiment. While keeping the FP on the same level as the baseline, our approach had a TP of 75.1% over 4029 testing tweets using Logistic Regression, a significant 5.4% improvement over the baseline.

III. METHODOLOGY

1. System Architecture

Presented in this section is the step-by-step process of the system. It is divided into 2 modules: (a) the pre-processing of the input and (b) sentiment analysis. Also, we used existing NLP (Natural Language Processing) tools to implement some of the methods included in the pre-processing module.

The system will only analyze the input if it is a proper English sentence and the words used are included in the WordNet 2.1. WordNet is a large database of English. Nouns, verbs adjectives are grouped into wets of cognitive synonyms (synsets), each expressing a distinct concept[15]. Censored words and abbreviations like FU are also prohibited to be included in the input. The analysis will be on sentence level. The input may either be a user input or file input.

1. Pre-Processing

For the sentence division, given an input the sentence divider method will divide the input, which can be a set of paragraphs, into sets of sentences. It is by detecting the ‘.’ (periods), ‘?’ (question marks), and ‘!’ (exclamation points) in the input. The next process is to spit the whole sentence into words and non-word elements like ‘,’ (commas), and ‘”’ (quotation marks). This is the tokenization process.

For the tagger and parser the research workers used the Stanford NLP Group POS Tagger and Stanford Parser. A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjectives, etc., although generally computational applications use more fine-grained POS tags like ‘noun-plural’. The Stanford POS Tagger software is a java implementation of the log-linear part-of-speech taggers in the papers [8] and [9] [10]. Each word will be classified if it is a person using the Name Entity Recognizer.

A natural language parser is a program that works out the grammatical structure of sentences, for instance which group of words go together (as “phrases”) and which words are the subject or object of a verb. The Stanford Parser package is a java implementation of probabilistic natural language parser, both highly optimized

*Figure 1, System Architecture*

PCFG and lexicalized dependency parser, and a lexicalized PCFG parser. The parser provides Stanford Dependencies output as well as phrase trees. The grammar that for the system is also from the Stanford Parser [11].

1. Sentiment Analysis

The sentiment analysis part of the system consists of definition extraction and sentence polarity analysis. For the definition extraction, the WordNet 2.1 and JAWS is used for getting the definitions based on the said application of each word. JAWS or Java API for WordNet Searching, as its name implies, it is an API that provides Java applications with the ability to retrieve data from the WordNet database. It is a simple and fast API that is compatible with both the 2.1 and 3.0 versions of the WordNet database files and can be used with Java 1.4 and later [12].

The Polarity analysis is done first by finding a bad word in the sentence. The part-of-speech of the bad word found in the sentence must be equal to the part of speech that has the bad definition. In determining if there is a bad context, there should be the receiver or object of an offense which is human resembling entity, most in a form of noun or pronoun. If the entity found is a pronoun it must not be in first person because there is no such thing as self-offense. If t is a noun, it should be a name of a person. If there is no word to act as the receiver of the bad word, then it will be considered as non-bad context state.

1. Testing and Experiment

The researchers compiled 100 comments with offensive words from different social media websites as the input for the testing and experimentation for this project. 50 comments were used for the testing and the other 50 for the experiment. For the experiment the researchers considered Table 1 and Table 2 for the experiment paper of this study.

Table 1 Each sentence is tested in system and by the expert

|  |  |  |
| --- | --- | --- |
| Sentence | Offensive | |
| System | Expert |
|  |  |  |
|  |  |  |
|  |  |  |

Table 1 is where the answers for the sentence analysis was recorded for each input. The expert labelled each input as yes, if the input is offensive, or no if it is not. The researchers also did the same for the system. Based from the results, each sentences was classified into the following:

* TP (True Positive) – expert and system both determined the input is offensive
* FP (False Positive) – System determined the input is offensive present, the expert indicated it’s not
* TN (True Negative) – both the expert and the system indicated that the input is not offensive
* FN (False Negative) – system indicated that the input is not offensive, the expert indicated it is offensive

Table 2 Input Scoring

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sentences | TP | FP | TN | FN |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Total: |  |  |  |  |
| Average: |  |  |  |  |

Table 2 is where each input will be labelled as true positive, false positive, false negative, and false positive. This scoring will be used for the evaluation of data in which the following metrics is used: Precision: the percent of identified comments that is offensive. Recall: the percent of offensive sentences correctly identified. Specificity: rate of the results without the condition, which have a negative test result. F- measure: the weighted harmonic mean of precision and recall. We used F- measure which gives equal weight to precision and recall.

1. Experiment Results

By using the 50 comments for experiment, an expert was asked to checked/tagged if the comments are offensive or not. The research workers gathered comments for the experimentation from a compilations of text file of *9gag.com* comments. These randomly chosen comments were used as input for the system. The following were the acquired results:

Table 3. Results from expert’s and system’s analysis

|  |  |  |
| --- | --- | --- |
| Sentences | Offensive | |
| System | Expert |
| Sentence 1 | No | No |
| Sentence 2 | No | No |
| Sentence 3 | No | No |
| Sentence 4 | No | No |
| Sentence 5 | No | No |
| Sentence 6 | No | No |
| Sentence 7 | No | No |
| Sentence 8 | No | Yes |
| Sentence 9 | Yes | Yes |
| Sentence 10 | No | No |
| Sentence 11 | No | No |
| Sentence 12 | No | No |
| Sentence 13 | No | No |
| Sentence 14 | No | Yes |
| Sentence 15 | No | No |
| Sentence 16 | No | No |
| Sentence 17 | No | No |
| Sentence 18 | No | No |
| Sentence 19 | No | No |
| Sentence 20 | No | No |
| Sentence 21 | No | No |
| Sentence 22 | No | Yes |
| Sentence 23 | No | No |
| Sentence 24 | No | No |
| Sentence 25 | No | No |
| Sentence 26 | no | No |
| Sentence 27 | No | No |
| Sentence 28 | No | No |
| Sentence 29 | No | No |
| Sentence 30 | Yes | No |
| Sentence 31 | No | No |
| Sentence 32 | No | No |
| Sentence 33 | Yes | Yes |
| Sentence 34 | No | No |
| Sentence 35 | No | No |
| Sentence 36 | No | Yes |
| Sentence 37 | No | No |
| Sentence 38 | Yes | No |
| Sentence 39 | No | No |
| Sentence 40 | No | Yes |
| Sentence 41 | No | No |
| Sentence 42 | Yes | No |
| Sentence 43 | No | No |
| Sentence 44 | No | No |
| Sentence 45 | No | no |
| Sentence 46 | No | No |
| Sentence 47 | No | No |
| Sentence 48 | Yes | Yes |
| Sentence 49 | Yes | Yes |
| Sentence 50 | Yes | Yes |

Table 3 shows the analysis for each sentences done by the system and the expert.

Table 4. Summation of the labels of the input

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | TP | FP | TN | FN |
| Total | 6 | 3 | 37 | 4 |
| Average | 1.714286 | 1.285714 | 6.142857 | 1.428571 |

In table 4, it shows the summation of all the sentences labelled as true positive, false positive, true negative, and false negative. The total no of the identified offensive sentences by the system and the expert is 6 while the total no of the identified sentences as not offensive is 37. The total no of the sentences labelled by the system as offensive but the expert indicated as not offensive is 3. While on the other hand, there are only 4 sentences in which the system identified as not offensive but contradicts the analysis of the expert.

Table 5. Performance Measure

|  |  |
| --- | --- |
| Specificity | 92.5% |
| Precision | 66.67% |
| Recall | 60% |
| Accuracy | 86% |
| F-measure | 63.1594% |

In table 5, the evaluated results is showed. The measure of precision and recall are 66.67% and 60% respectively. The results are still considered low mainly because the analysis for this type of evaluation is based from the true positive score. Referring to the results in table 4, the number of the comments identified as not offensive both by the system and the expert is 37 out of 50. Thus, the results for the recall, which measures the detection rate, the true positive rate, or the sensitivity rate of the system to identify offensive words is low but the measure for the specificity rate or the true negative rate, is 92.50% which is high. Since the measures for both the precision and recall is low, the weighted measure for the two is only 63.1594.

1. Conclusions, Recommendations, Further Works

In this study, the researchers develop a system that will analyze if a given input is offensive or not by using the definition of the offensive word present in the input as the basis for computation of the offensiveness of the sentence. The experiment has proven that the using the definitions as the basis of sentiment analysis is possible to give positive results. The utilization of relational inference analysis model of computation for the offensiveness is stable, if it has a narrow domain of discourse of analysis, which is an offense to a human entity.

For the recommendations, the system should be capable of disambiguating the word with the right definition during the extraction of meanings. The system should also have a larger domain of “Strong others” offensiveness, like offensiveness in non-person entities. The use of ‘not’ should also denote offensiveness of the sentence and not just a non-offensiveness keyword. Due to the informal structure of the sentences of the social media sites and forums, there should be a mechanism for the analysis of these sentences that has an improper structure, which will make it more suitable to apply in the specific domain.

# REFERENCES

[1] Dr Y Bi.*Analysing Social Media to Detect Cyber Bullying using Sentiment Mining.*School of Computing and Mathematics, Faculty of Computing and Engineering at the Jordanstown Campus of the University of Ulster. <http://www.findaphd.com/search/ProjectDetails.aspx?PJID=56055>

[2] Jun-Ming Xu, Xiaojin Zhu, Amy Bellmore(August 12 2012). *Fast Learning for Sentiment Analysis on Bullying*.Beijing,China:ACM. <http://pages.cs.wisc.edu/~jerryzhu/pub/wisdom12.pdf>

[3] Ying Chen, Yilu Zhou, Sencun Zhu HengXu .*Detecting Offensive Language in Social Media toProtect Adolescent Online Safety.*The Pennsylvania State University, University Park, PA, USA. George Washington University, Washington, DC,USA. http://faculty.ist.psu.edu/xu/papers/Chen\_etal\_SocialCom\_2012.pdf

[4] Dawei Yin, ZhenzhenXue, LiangjieHong(April 21, 2009).*Detection of Harassment on Web 2.0.* Madrid, Spain.<http://www.cse.lehigh.edu/~brian/pubs/2009/CAW2/harassment.pdf>

[6] ZhiXu, Sencun Zhu (n.d). Filtering Offensive Language in Online Communities using Grammatical Relations.<http://ceas.cc/2010/papers/Paper%2010.pdf>

[7] Guang Xiang, Bin Fan, Ling Wang, Jason Hong, Carolyn Rose. Detecting Offensive Tweets via Topical Feature Discovery over a Large Scale Twitter Corpus.<http://www.cs.cmu.edu/~lingwang/papers/sp250-xiang.pdf>

[8] Kristina Toutanova and Christopher D. Manning. 2000. [Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger](http://nlp.stanford.edu/~manning/papers/emnlp2000.pdf). In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), pp. 63-70.

[9] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. [Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network](http://nlp.stanford.edu/~manning/papers/tagging.pdf). In Proceedings of HLT-NAACL 2003, pp. 252-259.

[10] *Stanford Log-Linear Part-Of-Speech Tagger*. The Stanford Natural Language Processing Group.http://nlp.stanford.edu/software/tagger.shtml

[11] *The Stanford Parser: A statistical parser.*The Stanford Natural Language Processing Group.<http://nlp.stanford.edu/software/lex-parser.shtml>

[12] Java API for WordNet Searching (JAWS). <http://lyle.smu.edu/~tspell/jaws/>

[13] *WordNet: A lexical database for English* (August 26,2014).Princeton Univeristy.http://wordnet.princeton.edu/