Multi-level classifier for the detection of insults in social media

Reinald Kim Amplayo
Ateneo de Davao University
E. Jacinto St., Davao City 8000
Philippines
rktamplayo@addu.edu.ph

Jason Occidental
Ateneo de Davao University
E. Jacinto St., Davao City 8000
Philippines
jtoccidental@addu.edu.ph

ABSTRACT

Insults in social media websites create negative interactions within the network. These remarks build up a culture of disrespect in cyberspace. Automated insult detection methods are now used to regulate these posts. However, current implementations on insult detection using machine learning and natural language processing have very low recall rates. Thus, the study investigates the use of a multi-level classifier to predict insulting content. At its first level, a lexicon-based classifier is used to get the lexical score of the text. Then, at the second level, two Support Vector Machine (SVM) classifiers are used with word n-grams and character n-grams as input features. These classifiers output two numbers which are two classifications based on the word and character n-grams. At the final level, the results of the first two classifiers, combined with other input features, which include the number of characters, curse words, second person pronouns, capital letters, and symbols, are used as inputs for the neural network. The classifier has a 75.76% precision rate, 71.22% recall rate, 73.42% F1 score, and 86.71% AUC score. The multi-level classifier outperforms other methods in insult detection that use the same dataset and can provide a more reliable way to identify and omit insults in social media.

Categories and Subject Descriptors

I.2.7 Natural Language Processing: Text Analysis

General Terms

Design and Languages

Keywords

insult detection, multi-level classification, text classification, neural networks, support vector machines, social media

1. INTRODUCTION

Social media is rapidly growing every year. People of all ages are spending most of their time in the internet at social networking sites. They are able to interact with people despite the physical distance. Social media platforms can also be an avenue for people to share their opinions, give valuable insights, and learn new things. Applications and games are also made available in social media sites to increase patronage among users. A fact sheet provided by Pew Internet Project shows that as of January 2014, 74% of online adults use social networking sites. Majority of these users are 18-29 years old, which is 89% of the population [1].

People using these websites may come from different cultures and nationalities from around the world. Because of these cultural differences, there are often misunderstandings from user interactions. Misunderstanding may lead to conflict between users. Due to anonymity and freedom in social media, rude and insulting posts are easily made. Furthermore, some of these may even get famous and trend around the internet. These remarks highly affect the behavior of users when they are in the internet and when they are interacting with the real world. The users that are affected the most are the ones being insulted. They may get hurt and frustrated from the verbal abuse they received through social media. Consequently, they might respond to the insult with another insult. Thus, creates a negative interaction between users. These conditions make social media very vulnerable and fragile. 19% of teens reported that someone has actually written mean, offensive, and insulting things about them [2].

Because of this alarming issue, some organizations and websites provided solutions. The Terms of Service (ToS) already include a section regarding insulting posts, but people tend to disregard these policies when signing up. In forums and news groups, there is a "mark as inappropriate" button, but these features are often prone to collusions and are misused. In the Philippines, the Cybercrime Prevention Act of 2012 was approved to address legal issues concerning online interactions over the Internet, including cyber bullying. However, users of social networking sites from the Philippines still continue posting insulting remarks. Among common insults are addressed to public figures or companies.

Many researchers have tried to implement automatic intelligent software for detective insults. At present, pattern recognition and machine learning [3,4,5,6,7], and natural language processing [3,8,9,10,11,12] algorithms were used in automatically detecting insults. However, these implementations have low recall rates. In this study, a multi-level classifier is used to automate the detection of insults in social media. This involves three types of classifiers: a lexicon-based classifier that uses the lexical features; two n-gram SVM classifiers that used word and character n-grams; and the feed forward neural network classifier that integrates the frequency features and the results of other classifiers.

2. SOCIAL MEDIA INSULT DETECTION

An insult is an expression, statement or behavior that is considered to be offensive and impolite. Insults might be composed of racial slurs, profanity or other offensive language. In social media, posts with insulting or offensive language are frequently used for cyber bullying in online communities [13].

Insult detection is the process of predicting whether a text is insulting or not. This is a kind of text classification problem can be solved manually or automatically. Manual detection can be done by classifying each text intellectually by means of reading the texts. This process is very time-consuming and infeasible. Automatic detection of insults may use several machine learning and natural language processing techniques.

2.1 Current Approaches in Detecting Insults

A framework called Lexical Syntactic Feature (LSF) is used to detect offensive language in social media using two major features: lexical and syntactic features [8]. Lexical features treat each word and phrase as an entity. The lexicons used are profanities that are divided into two bag-of-words depending on their magnitude. The syntactic feature is an intensifier created using an assumption that if insulting words are directed to online users, they become more offensive. The framework creates combinations of a dependency-type word-pair form. This framework achieves high precision and recall rates, but sometimes misidentifies noun appositions. The framework did not take into account the possibility of typographical errors in social media texts. This creates a loss of correctness in grammar. The lexical features alone make the framework overfit. This resulted to a rather lower recall rate.

The series of experiments conducted by Vandersmissen [7] in the automated detection of offensive languages and behaviors in social networking sites include text classification based on lexicons and fuzzy algorithm, and on Naïve Bayes classifiers. The lexicon-based classifier contains list of profane words with the following properties: classification, intensity, centrality, and part-of-speech tag. These properties were manually crafted and adjusted based on experts' experiences and assessment skills. After the construction of the word list, a fuzzy algorithm is applied to every word of the input message and checks its intensity and centrality. The scores of these words are accumulated and would serve as the score of the message for a certain classification. The highest class score becomes the class of the message. The results have a very poor precision rate. The performance was improved by combining this classifier to a support vector machine (SVM) classifier which improved the precision rate.

A Naïve Bayes (NB) classifier is also included in the experiments in [7] but was later discarded in the final classifier build. The classifier used the term frequency – inverse document frequency (TF-IDF) model and was validated using two-fold cross validations. When presented with a new dataset, the classifier performed very poorly. The reason is due to the simplicity of the Naïve Bayes method. It cannot see and take into account the structure of a message clearly, which led to very low true positives.

The Naïve Bayes method is also used to detect peer-to-peer insults [4]. The method is compared with the logistic regression approach. In the NB method, it was assumed that the feature set is a multinomial probability distribution. Not all extracted features are used because only features that are pairwise independent can be used in a Naïve Bayes feature set. The logistic regression approach is used to address the independence issues. Although the logistic regression is sensitive to over-fitting, the evaluations conclude that the logistic regression is better than the NB method. Both methods can also be combined and is referred to as CNL. This new approach is also applied with Ada Boost and k-fold validations. The accuracy of the classifier did not improve.

A more successful implementation of the Naïve Bayes method is used by Razavi [6], which detects offensive language using multi-level Naïve Bayes classifier and an offensive language dictionary. The approach makes use of three levels of classification, all of which are kinds of Naïve Bayes classifier. The first level uses Complement Naïve Bayes classifier to select the most useful features out of the thousand features available. The second level uses Multinomial Updatable Naïve Bayes classifier. This level is to make sure that new training dataset can be used as this classifier is updatable. The last level is a rule-based classifier named DTNB (Decision Table/Naïve Bayes hybrid classifier). The last level is a combination of a decision table and a Naïve Bayes classifier. All three classifiers are found in the Weka package. As of the writing of this literature, there were no accurate datasets available to test the accuracy of the classifier.

2.2 Multi-level Classification

This study makes use of a multi-level classification framework shown in Figure 1.

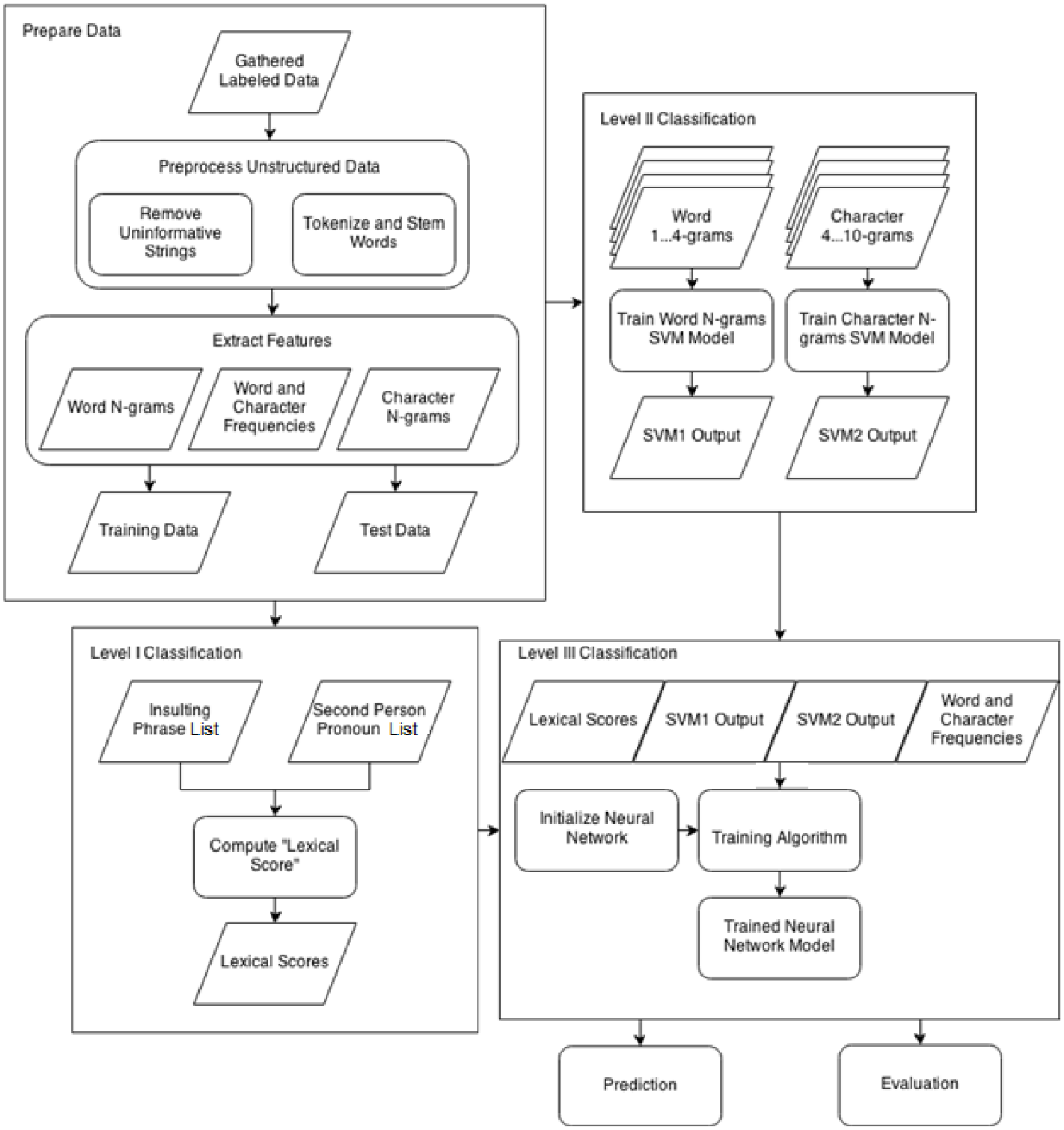


Figure 1. Multi-level classification conceptual framework.

After the preprocessing of data, the dataset is passed to two types of classifier: the lexicon-based classifier and the n-gram SVM classifier.

The lexicon-based classifier solely makes use of the lexical features of the given text. It does not use the dataset to learn with supervision. It uses word lists and calculates a lexical score based on those lists.

The n-gram SVM classifiers are used to reduce the dimension of the feature set. The feature sets include word n-gram frequency model and character n-gram frequency model. The frequency models include thousands of n-grams and feeding this directly to the neural network will make the network converge slowly and tend to over-fit. Each SVM classifier outputs one value, the predicted value based on the feature sets.

The lexical score and the outputs of the SVM classifier are combined with other frequency features as one feature set. This feature set becomes the input of the feed forward neural network. The network is trained using the dataset and will be ready for future data.

3. DATA

There are two kinds of data that needs to be gathered. The first kind is the training and test datasets. Impermium, a cyber-security startup that provides security for websites, released the datasets in Kaggle, a platform for data prediction competitions. The dataset only consists of three columns: the date when the text was made, the text itself, and the classification. In this dataset, insults are considered if they are intended to be insulting to a person who is a part of the larger blog/forum conversation. Insults could contain profane words, but often times, they do not. Texts which contain profane words but are not necessarily insulting are considered not insulting. There are 3947 rows in the training dataset while there are 4881 rows in the test and verification dataset combined. There is a small amount of noise in the labels as they have not been meticulously cleaned.

Table 1. Sample Dataset (Curse Words are Censored)

ID	Insult?	Date	Comment
1	1	20120618192155Z	"You **** your dad."
4	0	(blank)	"listen if you dont wanna get married to a man or a women DONT DO IT. what would it bother you if gay people got married stay in your lane do you let them do them. And your god is so nice but quick to judg if your not like him, thought you wasnt suppose to judge people."
9	1	20120502173553Z	"Either you are fake or extremely stupid...maybe both..."
16	1	20120611090207Z	"FOR SOME REASON U SOUND ********. LOL. ****. WHERE U BEEN ****"

Two word lists are also needed for the lexicon-based classifier. These are the curse word list and the second-person pronoun list. The curse word list is obviously necessary since texts that have lexicons used for cursing tend to be insulting. The second-person pronoun list assures that the curse words are directed to a person who is a part of the conversation. There are 1048 curse words gathered from multiple sources. The second-person pronoun list consists only of eight words. These are you, your, ya, u, ur, yourself, yo, and yours.

4. LEXICON BASED CLASSIFIER

The lexicon-based classifier assumes that if there are curse words and second-person pronouns on the same text, the text is more likely to be insulting. It further assumes that the closer they are in the text, the more likely the text to be insulting. A pseudo-code shown below is created using these assumptions to calculate the lexical score.

```
FOREACH text in dataset
    GET stemmed words in text AS stemmed
    SET lexical_score to 0
    IF stemmed has curse words AND
      stemmed has "you" words THEN
        GET all indices of curse words
          AS curse_words
        GET all indices of you words as you_words
        FOREACH curse_pos in curse_words
            FOREACH you_pos in you_words
                SET lexical_score to
                  (lexical_score +
                  ABS(curse_pos - you_pos))
    GET lexical_score
```

Figure 2. Pseudo-code to get the lexical score.

The lexical scores are normalized to fit into the [0, 1] range. The precision and recall rates were 100% and 66.1%, respectively. The F1 score was 79.59% and the area under the curve (AUC) score was 76.94%. The reason behind the perfect precision is that it classifies text with a curse word and a second person pronoun as insulting. In the test and verification datasets, all the texts that were classified as insulting have at least one curse word and second person pronoun.

5. N-GRAM SVM CLASSIFIERS

Before creating the n-gram feature vectors, the text needs to be preprocessed. The basic preprocessing necessary is to erase the Unicode texts and symbols that were not created by the user. A simple named entity recognizer is used to classify the named entities and unique symbols such as:

- Email addresses
- URLs
- Twitter usernames
- Curse-like symbols (such as *!&@#^)
- Repeated question marks
- Repeated periods
- Smileys
- Repeated letters in a word (such as wooooord)

The minimum word n-grams are unigrams and the maximum word n-grams are 4-grams. The minimum character n-grams are 4-grams and the maximum character n-grams are 10-grams. The support vector machine is run with a linear kernel implemented in terms of the LiblineaR library. The classifier has more flexibility in the choice of penalties and loss functions and scales better in large numbers of samples.

After normalizing the SVM classifier results to fit in the [0,1] range. The precision and recall rates for the character n-gram SVM classifier were 71.36% and 68.79%, respectively. The F1 score and AUC score were 70.05% and 84.39%, respectively. The word n-gram SVM classifier has a slightly higher precision with 71.75%, but has a lower recall which was 65.73%. This reflects the lower F1 and AUC scores, which were 68.61% and 83.17%, respectively.

6. FEEDFORWARD NEURAL NETWORK

The neural network accepts 10 inputs. The first three inputs are the results of the lexicon-based classifier and the SVM classifiers. Seven of them are frequency features from the text. These are:

- Number of curse words
- Number of second-person pronouns
- Number of characters
- Number of exclamation points

- Number of question marks
- Number of asterisks
- Number of capital letters

The neural network has one hidden layer with 15 nodes. The learning rate is 5 x 10^{-6} and the momentum is 1 x 10^{-1}. The learning rate was set to a very low value since the variance of all the features was relatively small. The Sigmoid function was used as the activation function for both the hidden and output layers. After 214,037 iterations, an error rate of 1 x 10^{-3} was achieved.

7. RESULTS

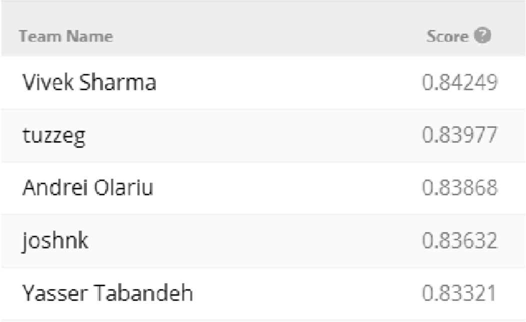
Three types of evaluation metrics were used to compare performance of different types of classifier. These were the precision and recall rates, the F1 score, and the area under the curve (AUC) score. Table 2 shows the evaluation metrics of different types of classifiers. The best scores are bold-faced, the below average scores are italicized, and the worst scores are bold-faced and highlighted gray.

Table 2. Evaluation of Different Classifiers

Level	Frequency	Character SVM	Word SVM	Lexical Score	Precision	Recall	F1 score	AUC
1				✓	**100**	*66.1*	**79.59**	***76.94***
1	✓				76.33	***58.82***	***66.44***	*80.79*
1			✓		*71.75*	*65.73*	*68.61*	*83.17*
1		✓			*71.36*	68.79	*70.05*	*84.39*
1		✓	✓		77.57	*65.13*	*70.81*	84.81
2	✓			✓	***70.73***	*67.71*	*69.19*	*83.19*
2			✓	✓	*73.9*	68.2	*70.93*	*84.7*
2	✓		✓		*71.64*	69.21	*70.41*	*84.78*
2	✓	✓	✓		79.38	*65.44*	71.74	85.6
2		✓		✓	77.85	*66.8*	71.9	85.73
2	✓	✓			*75.54*	68.53	71.86	85.87
2		✓	✓	✓	76.1	68.45	72.07	85.97
3	✓		✓	✓	*73.95*	68.61	71.18	85.18
3	✓	✓		✓	76.78	68.71	72.52	86.19
3	✓	✓	✓	✓	75.76	**71.22**	73.42	**86.71**

The final multi-level classifier, a three-level classifier that uses all the input features, has the best recall rate and AUC score. It also has an above average F1 score and an average precision rate. Compared to the other classifiers, the final multi-level classifier performs relatively better than the other classifiers.

A competition was held by Impermium at Kaggle last September 2012 for insult detection in social commentary. The leaderboard contains the AUC scores of the classifiers of different people. In Figure 3, the top five teams and their respective AUC scores are shown. It is evident that the AUC score achieved by the final multi-level classifier is higher than that of the first place in the competition.



Team Name	Score
Vivek Sharma	0.84249
tuzzeg	0.83977
Andrei Olariu	0.83868
joshnk	0.83632
Yasser Tabandeh	0.83321

Figure 3. Kaggle contest leaderboard screenshot.

8. Conclusion

The study presented a multi-level classification approach to the detection of insults in social media. A feed forward neural network was trained with the use of a lexicon-based classifier, two n-gram SVM classifiers, and seven frequency features from the dataset. The results show an improvement in the evaluation scores as the number of levels increase. Therefore, it is really necessary to have multiple levels of classification for this type of problem.

It is also shown in Table 2 that the multi-level classification has better recall rates as compared to a single classification technique. This solves the problem on previous research papers concerning the low recall rates they get.

The results were also compared to the leaderboard of the Kaggle contest where the dataset used in this paper was acquired. It is apparent that the difference between the first place of the competition and the classifier in this paper is approximately 2.46%.

In the Kaggle competition, some of the contestants shared their experiences in a discussion thread [14]. Majority of their applications used linear classifiers word n-grams, and character n-grams. Vivek Sharma, the winner also shared that the use of SVM classifiers worked better in his solutions over linear regression.

Thus, the use of a multi-level classifier for insult detection in social media was proven to outperform the existing techniques available as well as solutions presented from the kaggle programming contest.

9. ACKNOWLEDGEMENTS

This study would not have been possible without the support of many people. The authors wish to express their gratitude to their coordinator, Prof. Grace Tacadao. The authors would also like to express their gratitude to Professors Michelle Banawan and Dominique Cimafranca for their advice and support for this research. The authors would also like to acknowledge Impermium who made and provided the dataset used in this study. Finally, the authors wish to thank their loving friends and family.

10. REFERENCES

[1] Pew Research Internet Project. (n.d.). Social Networking Fact Sheet. Available July 31, 2014: http://www.pewinternet.org/fact-sheets/social-networking-fact-sheet/

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

[3] Adler, B., De Alfaro, L., Santiago M., Rosso, P, West, A., Wikipedia vandalism detection: combining natural language, metadata, and reputation features, Proceedings of the 12th international conference on Computational linguistics and intelligent text processing, February 20-26, 2011, Tokyo, Japan

[4] Decoster, B., Ibrahima, F., and Yang, J. Detecting peer-to-peer insults. in CS 229 Machine Learning Final Projects, (Stanford University, 2012), ICME.

[5] Greevy, E. and Smeaton, A., Classifying racist texts using a support vector machine, Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, July 25-29, 2004, Sheffield, United Kingdom

[6] Razavi, A., Inkpen, D., Uritsky, S., and Matwin, S. Offensive language detection using multi-level classification. in Proceedings of the 23rd Canadian conference on Advances in Artificial Intelligence, (Berlin, Heidelberg, 2010), Springer-Verlag, 16-27.

[7] Vandersmissen, B. Automated detection of offensive language behavior on social networking sites. Master's thesis, Universiteit Gent, 2012.

[8] Chen, Y., Zhu, S., Zhou, Y., and Xu, H. Detecting offensive language in social media to protect adolescent online safety. in International Conference on and 2012 International Conference on Social Computing, (SocialCom, 2012), Privacy, Security, Risk and Trust, 71-80.

[9] Goyal, P. and Kalra, G., G.S. Peer-to-Peer Insult Detection in Online Forums, Indian Institute of Technology Kanpur, 2013

[10] Hong, J., Detecting Offensive Tweets via Tropical Feature Discovery, 2012

[11] Mahmud, A., Ahmed, K. Z., and Khan, M.. Detecting flames and insults in text. In Proceedings of the Sixth International Conference on Natural Language Processing, 2008.

[12] Spertus, E., Smokey: Automatic recognition of hostile messages, Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence, p.1058-1065, July 27-31, 1997, Providence, Rhode Island

[13] Xu, Z. and Zhu, S. Filtering offensive language in online communities using grammatical relations. In Proceedings of the Seventh Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference, 2010.

[14] Sharma, V. (2012, September 24) What did you use in detecting insults in social commentary [Msg 4]. Message posted to https://www.kaggle.com/c/detecting-insults-in-social-commentary/forums/t/2744/what-did-you-use