Cyberbullying Detection for Online Games Chat Logs using Deep Learning

Conference Paper · November 2019

DOI: 10.1109/HNICEM48295.2019.9072811

Bernie Fabito
National University, Philippines
30 PUBLICATIONS **124** CITATIONS
SEE PROFILE

Ramon Rodriguez
National University, Philippines
18 PUBLICATIONS **40** CITATIONS
SEE PROFILE

Manolito Octaviano
University of the Philippines
10 PUBLICATIONS **26** CITATIONS
SEE PROFILE

Nathaniel Oco
28 PUBLICATIONS **101** CITATIONS
SEE PROFILE

Cyberbullying Detection for Online Games Chat Logs using Deep Learning

James Albert Cornel, Carl Christian Pablo, Jan Arnold Marzan, Vince Julius Mercado, Bernie Fabito, Ramon Rodriguez,
Manolito Octaviano Jr., Nathaniel Oco, Angelica De La Cruz
College of Computing and Information Technologies
National University – Manila
Manila, Philippines
{corneljac, lacandazoccp, marzanjh, mercadovjh}@students.national-u.edu.ph
{bsfabito, rlrodriguez, mvoctavianojr, naoco, ahdelacruz}@national-u.edu.ph

***Abstract* — Cyberbullying is a form of harassment that takes place in the internet where a bully sends a harsh message to harass the receiver. In this study, a learning model is developed using Convolutional Neural Network (CNN), which is usually used for image, and is then used to create a system for detecting cyberbullying in online game chat logs. Chat logs collected from Dota and Ragnarok were preprocessed and annotated. 60% of the data were used for training, 30% were used for testing the generated model and the remaining 10% were used for validating the developed model. After validation, the accuracy of the CNN model yielded to 99.93%. Based on the results of validation, the CNN model tends to overfit, even with regularizations applied, and was not able to generalize well. For comparison, another model was created using Naive Bayes and the accuracy yielded to 92.23%. It can be concluded that detecting cyberbullying using Naive Bayes is already possible, however, the accuracy is still not comparable to existing DNN models. While the use of CNN model results to overfitting, it is recommended to explore on other DNN architectures for online games chat logs.**

Keywords — deep learning, convolutional neural network, cyberbullying detection, word embedding, normalization, online gaming

I. INTRODUCTION

Technology advancement has grown rapidly over the past decades, especially in online games which allowed billions of people to interact with each other. Online games bring enjoyment to gamers globally. However, there is also an opportunity for cyberbullying. This happens when a player makes a mistake in-game, the other players make a big deal of the mistake. Another instance is when a player does not play well, s/he is likely to be bombarded with abuse.

A survey conducted by Stairway Foundation Inc. in 2015 showed that 80% of young teens in the Philippines aged 13 to 16 have been victims of cyberbullying, and 60% of their child counterparts from age bracket of 7 to 12 were also cyberbullied. From this survey 30% of children from 7 to 12 and 40% from age 13 to 16 were aware that other children were also victim of cyberbullying. On the survey, 5 out of 10 aged 7 to 12 are okay with letting their parents know while 4 out of 10 aged 13 to 16 are okay with letting their parents know of their online activities. In terms of children, 60% of them would get help from their parents while only 34% from young teens would consult their parents, and 70% of children's parents are aware of their online activities while 50% of parent of teens. With this information, the need for automatic means of detecting cyberbullying arises.

Cyberbullying can take place in any platforms in the internet as long as there are interactions between multiple users. This situation is very evident on social media platforms such as Twitter, Facebook and Youtube, as well as in online games chat logs. However, most of the existing studies with regards to cyberbullying detection focuses on developing learning models for social media platforms [1]. In the recent studies, the use of deep neural networks (DNN) for cyberbullying detection outperformed the learning models created using machine learning (ML) algorithms.

The goal of this study is to develop a learning model for cyberbullying detection in Filipino online games chat logs by exploring the use of convolutional neural network architecture. This study will aid future researchers in developing a system that would block the detected chat with bullying content in online games which can help prevent further harassment. This study will also reduce manual monitoring efforts online games, and may play a role in preventing teenagers to suffer from further depression and low self-esteem due to cyberbullying. As of this time, there are no recent studies for cyberbullying detection in online game chat on the Filipino community of gamers.

II. RELATED WORKS

Traditional machine learning methods are the most widely used approach in detecting cyberbullying. These methods were Support Vector Machine (SVM), Naïve Bayes, Nearest Neighbor, and Decision Trees. In the study of Haidar et.al. [2] Naive Bayes and SVM were used to detect cyberbullying in Arabic content. The study was able to achieve good results. Naïve Bayes had an overall F-Measure of 90.5% while SVM having the highest overall F-Measure of 92.7%. However, the problem in this study was the high percentage of false positive. The use of decision tree was utilized on the study of Reynolds et.al. [3] which produced a language-based method in detecting cyberbullying that detected a small sample of data from Formspring.me using J48 decision tree which out-performed other used algorithms in his study with 78.5% accuracy. The data of the study contained 18,554 users with posts of 1 to 1000 for every user.

Another study by [4] used Vector Space model that uses algebraic approach of filtering data. The study also used J48 as one of their algorithms but was out-performed by SMO-PolyKernel. Performing 10-fold Cross-validation which

978-1-7281-3044-6/19/$31.00 ©2019 IEEE.

gave a result of 68.47% accuracy but lower compared to the previous study since the language is in Spanish and its culture of cyberbullying differs.

In the study of [5], the researchers proved that the use of deep learning outperformed the traditional machine learning models. The CNN having the highest F-score of 0.91, while LSTM had 0.88, BLSTM had 0.88 and lastly BLSTM with attention had 0.90. In order to achieve this result, the researchers oversampled the bullying labeled data thrice and evaluated the results using five-fold cross validation.

Most the studies only focused in detecting cyberbullying in social media platforms and few used game chats. This study will be focused in detecting cyberbullying using game chats from the Filipino community of gamers.

III. Methodology

In this chapter, the researchers showed the step by step procedure in detecting cyberbullying using various methods, from preprocessing to final analysis. The system methodology is shown in Fig. 1.

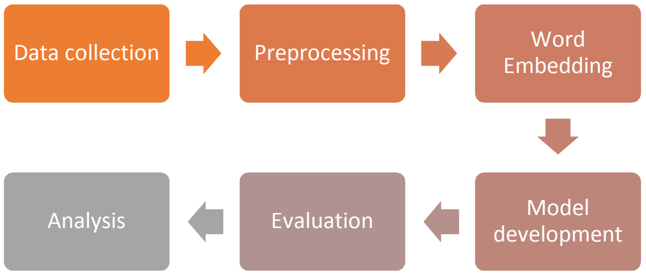

Fig. 1. System methodology for detecting cyber bullying

A. Data Collection

The researchers used the data provided by [6], which were gathered from an MMORPG game in the philippines which is ragnarok. It was collected using OpenKore – a free open source client + bot program made specifically for ragnarok. Additional data were collected from two different servers of DotA Japan and Singapore server, and iPlay server Ragnarök. The researchers used opendota.com to gather data in DotA, using web API. The opendota.com is an open source website which provides free access to information such as matches, player profile and chats.

The chats in ragnarok were classified into four: PM or personal message, C or public chat, and GM and S which can be read by everyone in the game. In total, the researchers successfully gathered 230,394 lines of phrases in DotA which were logs containing public chats form public and rank matches, and 534,328 lines of phrases in Ragnarök from public chats in the server. The total lines of phrases collected is 764,722. The time frames of the collected data were 2012, 2013 and 2019.

B. Preprocessing

All the data were transformed to lower case. The data that were gathered in Ragnarök contained number of unnecessary lines of chats like game announcements or game reports. A python script was used to remove the game announcement and reports by the researchers. To identify which line to be removed, the script looked at string [s], [gm] and shouts. If any string [s] was found or string [gm] and no shouts was found, then this line was removed.

The game reports of the Dota was different with the Ragnarök, it only contained numbers that is why the cleaning process for the data that was gathered in Dota was different with the cleaning process for Ragnarök. In Dota the researchers split the phrases by (,) and obtained the first index that contained the chat of the players. All the game reports of the Dota were removed after the removal of stopwords since it only contained numbers, the end result of the game reports was empty. Sample chats that were removed from the dataset are shown in Table 1.

TABLE I. Sample Chats Removed from the Dataset

Game	Message
Ragnarok	[Jan 25 17:29:10 2013][S] Congratulations! ~vi/\/~ just received '+8 Weapon Refine Deed' from Lotti Gurl (prontera 142 228)!
Ragnarok	[Mar 12 14:19:38 2013][S] Tentacle ni Uncle shouts: Hahaha Gago. Shout ka pa sige, dami nakakakilala sayo LOWLIFE
Dota	2,0,0,2118,chatwheel,
Dota	napakaingay mo talaga,130,7,2412,chat,Stray GOD ?

From 764,722 lines of phrases, the researchers successfully removed 169,045 lines of game reports and game announcements from the game Ragnarök and Dota, leaving the researchers 525,598 lines of phrases.

The data also underwent normalization, 164 unique words were normalized manually using Notepad ++. A total of 266,596 words were affected with the normalization process. Normalization was done because the collected data contains words that are the same but are spelled differently. These words may affect the performance of the model. Special characters and stop words were not needed because they did not have enough descriptive power to distinguish between relevant and not relevant in the data. That is why stop words and special characters were also removed by the researchers in order to improve the performance of the model [7].

- Some stop words that were removed: akin, aking, ako, alin, am, sa, sa, na, amin, aming, ang, ano, anumang, apat, at, atin, ating, ay, bababa, bago, bakit, bawat, bilang, dahil, dalawa, dapat.

- Special Characters that were cleaned: [ ] @ < > ( ) ! { } ! = -_ / ~ + | , : ; $ # ^ * & ? .

The data were softly labeled using scoring system. Bullying keywords and negation keywords were counted. The counted bullying keywords were subtracted from the counted negation keyword. If the result is greater than zero, then the phrase is labeled as bullying otherwise not-bullying.

From 525,598 lines of phrases, 20,355 lines of phrases were labeled as bullying and 499,718 phrases as not-bullying. In order to avoid class imbalance, the researchers only used 20,355 lines of bullying phrases and 20,355 lines of not-bullying phrases. From the total 40,710 phrases, 60% were used for training, 30% for testing and 10% for validation. Five gamers were used to validate the labeled phrases. 81.47% were correctly classified and 18.53% were incorrectly classified.

C. *Word Embedding*

Word embeddings are vector representations of a word which are mapped to vectors of numbers in order to give meaning to a certain sentence.

These word vectors represented words as multidimensional continuous numbers which had closeness or similarity in value (see Fig. 2). Example is the word bobo which is a synonym of tanga which is shown in the vector that they are similarly correlated based on the closeness of each vector.

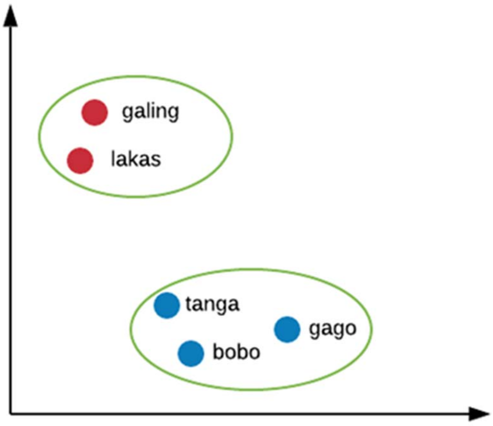

Fig. 2: Visual Representation of Word Vectors

D. *Convolutional Neural Network*

CNN was used to classify the binary classes: bullying and non-bullying chat logs. Although CNN has been used for image processing, it has been also used effectively for text classification [8, 9]. Since text is a Sequential data, one dimensional convolution was used. The parameters used in the neural network were tuned based on a series of experiments. Initial parameters were set randomly and were then tuned to get the hyper parameters. See Fig. 3 for the architecture used in the study. The word embeddings with a dimension of 50 were used as a feature of the CNN model which obtained inputs and concatenates it into subsets then passed it to the convolutional layer with three layer that performed the convolution operation and generated a feature map that created new features. These feature maps were the patterns of words that make a certain meaning.

The pooling layer simplified the output of the convolutional layers by reducing the size of the output compiling each feature into one vector. The used pooling technique was max-pooling which filtered out unneeded composition of words [8]. From the sack of layers, different pooling techniques were while preserving important features to be sent to the final layer or the fully connected layer which filtered the new features classified the sentence [9].

The model was optimized by using dropout which shuts down some neurons in a neural network which were chosen by random which helps in learning more robust features. The value used in the dropout layer is 0.50. L1 L2 Regularization where L1 minimizes the less important features to zero and important features to non-zero weights and L2 only minimizes the weights [5, 8, 9]. The use of these regularizers and dropout layers is to avoid overfitting which the model performance from unseen data [10]. The value in the L1 regularizer is 0.0005 and in L2 regularizer 0.0001.

E. *Evaluation*

The researchers used the confusion matrix in order to look at the relative distribution of the classes to verify if the model correctly classified its true positives and true negatives. The accuracy of the matrix could be computed by computing the average number of values by the total number of true positives plus true negatives.

The evaluation used was the F1 Score of the model and was also used to measure the accuracy of the test accuracy which give the harmonic mean between the precision and recall. It showed the precision of the classifier and the robustness of the model. High precision but low recall is an accurate but misses instances that are difficult to classify. The higher the F1 score, the better the performance of the model.

IV. Results and Discussion

The researchers developed four models to detect cyberbullying for online game chat logs using Convolutional Neural network. All the models performed with a very high accuracy which suggests overfitting. During experimentation, the parameters that yielded with the highest validation accuracy is Experiment 3 which achieved a 99.86% validation accuracy. Experiment 4 yielded the lowest validation accuracy of 99.82%. See Table 2 for the complete results of the experiments. The four CNN models developed were compared with Naïve Bayes with four

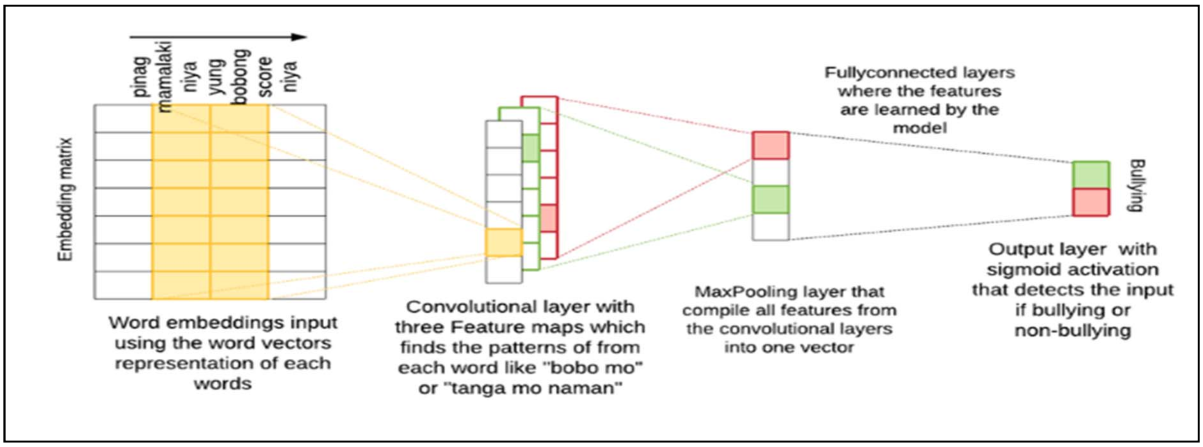
pinag
mamalaki
niya
yung
bobong
score
niya
Embedding matrix
Word embeddings input using the word vectors representation of each words
Convolutional layer with three Feature maps which finds the patterns of from each word like "bobo mo" or "tanga mo naman"
MaxPooling layer that compile all features from the convolutional layers into one vector
Fullyconnected layers where the features are learned by the model
Bullying
Output layer with sigmoid activation that detects the input if bullying or non-bullying

Fig. 3. Convolutional Neural Network Architecture

different features: Count Vectors, WordLevel TF-IDF, N-Gram Vectors, and CharLevel Vectors. 5-fold cross validation was used to compare with the testing accuracy. These also used Bag-of-words which was comparable to the deep neural network model used in the experiment.

To further validate the CNN model, it was compared to Naïve Bayes, a traditional machine learning algorithm. Just by looking at the results of the models of Naïve Bayes, it can be concluded that the CNN model used for the experiment suffered from over fitting. See Table 3 for the results.

TABLE II. EVALUATION RESULTS OF CNN MODELS

Experiment	Acc	Prec	Rec	F1
1	99.86	99.72	100	99.88
2	99.86	99.72	100	99.88
3	99.93	99.86	100	99.92
4	99.82	99.64	100	99.82

TABLE III. EVALUATION RESULTS OF THE NAÏVE BAYES MODEL

Feature vectors	Val Acc	Acc	Prec	Rec
CountVectors	89.02	89.44	90.86	89.37
TF-IDF	91.97	92.32	92.74	92.32
N-Gram	74.44	74.87	81.59	74.87
CharLevel	89.05	89.89	89.34	89.89

With the Naïve Bayes algorithm used for training the model, WordLevel TF-IDF as features achieved the best test accuracy of 92.32% having a precision of 92.74%, Recall 92.32%, and an F1-Score of 92.30% and with an accuracy of 91.97% performing 5-fold cross validation. Among all the features the N-Gram Vectors yielded the lowest validation accuracy of 74.87%, Precision 81.59%, Recall 74.87%, and an F1-Score of 73.49% with a validation accuracy of 74.44%.

By looking at the confusion matrix of the CNN model (see Table 4), it could be understood that there are no misclassified bullying phrases and only a handful of misclassified not-bullying phrases. On the other hand, the Naïve Bayes algorithm showed realistic results as shown in Table 5. In the Count Vectors model, prediction of bullying was unsuccessful. The model misclassified too many phrases of not-bullying as bullying. Word Level TF-IDF and Character Level Vectors performed well in predicting the phrases with Word Level TF-IDF having the best result. N-Gram Vectors had the lowest accuracy. The number of misclassified bullying words were nearly half of the class which the model failed to identify bullying phrases.

TABLE IV. CONFUSION MATRIX OF THE CNN MODEL

Experiment		Not-bullying	Bullying
1&2	Not-Bullying	1417	4
	Bullying	0	1429
3	Not-Bullying	1419	2
	Bullying	0	1429
4	Not-Bullying	1416	5
	Bullying	0	1429

TABLE V. CONFUSION MATRIX OF THE NAÏVE BAYES MODEL

Feature vectors		Not-bullying	Bullying
Count Vectors	Not-Bullying	4867	1215
	Bullying	75	6056
TF-IDF	Not-Bullying	5310	772
	Bullying	166	5965
N-Gram	Not-Bullying	5953	129
	Bullying	2940	3191
CharLevel	Not-Bullying	5075	1007
	Bullying	350	5781

The results of Convolutional Neural Network is higher than the results of Naïve Bayes. Based on the observation of the results, the CNN model suffered from overfitting. The CNN model is memorizing the data, but is not learning the bullying phrases and is simply becoming rule based learning that if it detects a bullying keyword included in the training dataset, it predicts the phrases as bullying and also if the word is out of vocabulary, it fails to correctly classify the phrase that is why it was compared to Naïve Bayes. In Naïve Bayes, it predicts participation probabilities for each class, for example, the likelihood that given record or information point has a place with a specific class.

In a study [11], it was stated that neural networks approach works well with short sentences yet experiences issues with long sentences and especially with sentences that are longer than those used for training. The results of text classification may be influenced by the length of the sentence in the data. Another study [12] stated that CNN can take long computational cost especially with the size of the sentence. The effect of longer sentences comes with more computation having longer training time.

The test accuracy of the Naïve Bayes model showed lower accuracy than the training which means it was a good fit. Comparing it to the validation of the CNN models that showed overfitting.

V. CONCLUSION AND FUTURE WORKS

CNN models were successfully developed for detecting cyberbullying in Filipino online games chat logs. The CNN model was able to achieve a validation accuracy of 99.92%. It can be observed that the model suffered from overfitting especially when it is asked to classify new words that is out of its vocabulary (words that are not included in the training set). It can be concluded that detecting cyberbullying using Naive Bayes is already possible, however, the accuracy is still not comparable to existing DNN models. For future work, it is recommended to explore on other DNN architectures for online games chat logs. Since the researchers were only able to normalize 164 unique of words. It is suggested to normalize more words in the data in order for the model to learn better.

ACKNOWLEDGEMENT

The researchers would like to give thanks to Mr. Joseph Marvin Imperial for his insightful recommendations on this study. We would also like to acknowledge the support provided by the College of Computing and Information Technologies through the Human Language Technology Laboratory.

REFERENCES

[1] Dadvar, M., & Eckert, K. (2018). Cyberbullying Detection in Social Networks Using Deep Learning Based Models; A Reproducibility Study. *arXiv preprint arXiv:1812.08046.*

[2] Haidar, B., Chamoun, M., & Serhrouchni, A. (2017, October). Multilingual cyberbullying detection system: Detecting cyberbullying in Arabic content. In Cyber Security in Networking Conference (CSNet), 2017 1st (pp. 1-8). IEEE.

[3] Reynolds, K., Kontostathis, A., & Edwards, L. (2011). Using machine learning to detect cyberbullying. In Machine learning and applications and workshops (ICMLA), 2011 10th International Conference on (Vol. 2, pp. 241-244). IEEE

[4] Galán-García, P., Puerta, J. G. D. L., Gómez, C. L., Santos, I., & Bringas, P. G. (2016). Supervised machine learning for the detection of troll profiles in twitter social network: Application to a real case of cyberbullying. Logic Journal of the IGPL, 24(1), 42-53.

[5] Agrawal, S., & Awekar, A. (2018, March). Deep Learning for Detecting Cyberbullying Across Multiple Social Media Platforms. In European Conference on Information Retrieval (pp. 141-153). Springer, Cham.

[6] Oco N., Syliongka, L. R., Allman T., Roxas R. T. (2016), Resources for Philippine Languages: Collection, Annotation, and Modeling, Graduate Institute of Applied Linguistics

[7] Cheng, C., & Ng, A. (2016). Automated Role Detection in Cyberbullying Incidents, Proceedings of the 16th Philippine Computing Science Congress, Puerto Princesa, Palawan, Philippines, March 16 – 18, 2016. Quezon City, Metro Manila, Philippines: Computing Society of the Philippines

[8] Hu, B., Lu, Z., Li, H., & Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In Advances in neural information processing systems (pp. 2042-2050).

[9] Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

[10] Overfitting and Underfitting With Machine Learning Algorithms by Jason Brownlee, Retrieved April 10, 2019 from: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/?fbclid=IwAR3IU1l8hUtQCxCKm2EsBEz_jfagrY2U2Wg93ucnskc43i6YcYEd0PVlD6E

[11] Pouget-Abadie, J., Bahdanau, D., Van Merrienboer, B., Cho, K., & Bengio, Y. (2014). Overcoming the curse of sentence length for neural machine translation using automatic segmentation. arXiv preprint arXiv:1409.1257.

[12] Song, X., Petrak, J., & Roberts, A. (2018). A Deep Neural Network Sentence Level Classification Method with Context Information. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 900-904).