UDC 004.912

Elton M. Dube., *4th year student,*
аспірант 1 курсу Specialty 122 "Computer Science"
Serhiy Shtovba, *professor,*
Computer Science and
Information Technology Department

HIDDEN BAD WORDS IDENTIFICATION IN SOCIAL NETWORK CONTENT

Vasyl' Stus' Donetsk National University, Vinnitsia

With the increase in popularity and availability of different social media platforms, more and more people are finding it easier and easier to communicate with each other all over the world. That is just one of the highlights of social media. Now let us look at the downside of social media. While many people use it for simple communication with friends and family and keeping up with the latest trends and news, others on the other hand are using it for all the wrong reasons including bullying and circulating false information. All of this has led to the development and improvements of things such as the detection of abusive language, hate speech, cyberbullying, and trolling amongst others. Social Media Sites are being tasked to continuously improve their cybersecurity measures to protect their users from cyberbullying.

With the world ever evolving and humans becoming more and more sophisticated and intelligent, this has led to cyberbullies adapting to these restrictions put in place by social media sites and now masking bad words, profanity and hate speech, and this has made it hard for some models to detect some bad words and profanity.

With the help of machine learning we have come up with ways to detect masked bad words in social media content and in this report, we will look at some models that are being used and how we can further improve them in the future.

Some examples of masked bad words that are hard to detect automatically include the following: *b1tch, 5hit, A55, D!ck, fvck, cr@p, D1ps#!t, pr1ck, idi0t, idl0t, Pus*y*. All these words are out of vocabulary, but a human recognizes easily.

With more room for improvement, we have seen a few models/techniques being developed over time. In 2012 Sara Owsley Sood and other co-authors [1], developed a technique that used Crowdsourcing as its main model. This model used useful features like Bigrams, but this model had a very low recall performance, and the system could not be optimized for a high recall performance.

In 2019 we saw the development of a model that used Sentence embeddings from Vijayasaradhi Indurthi and other co-authors [2], and this model used word embeddings and sentence embedding as main features. One drawback to this model is that the class distribution was highly imbalanced due to which there was a likelihood of a bias being introduced by the training algorithms.

Another model that was proposed in 2019 was Perspective & Bert by John Pavlopoulos and other co-authors [3], and This model made use of character n-grams, word-length distribution, extra-linguistic features and geographic features. Although good it had a few drawback and one was that the geographic and word length distribution have little to no positive effect on performance and rarely improve over character-level features.

A Deep machine learning model was proposed by D. Thenmozhi and other co-authors [4]. This model made use of Word embeddings, Multinominal Naïve Bayes, SVM, Stochastic Gradient Descent, Bag of words, Bi-gram features, Skip-grams, clustering-based word representations. The drawback observed from the results of this model was that the deep learning model could not learn the features appropriately due to less domain knowledge imparted by the smaller dataset used.

In 2020 a Traditional Machine Learning Model was proposed by Varsha Pathak and co-authors [5]. This model saw the use of features like Word n-gram, character n-gram, combined word, custom word embedding. Drawback to this model that we observed was that it cannot learn offensive terms from the text contents or from speech irrespective of the language.

Our approach to detecting the masked bad words is as follows. The first task at hand will be to source out a vocabulary of known bad words in English that we can use for comparisons later. For this we found the dataset “Bad Bad Words” on Kaggle, and the purpose of this dataset is to support the Toxic Comment Classification Competition. It has a wide range of words, i.e., close to 2000 words.

Now in order to analyze the data we will use embedding algorithms like Word2Vec because it is a statistical method for efficiently learning a standalone word embedding from a text corpus as there is no need to analyze a full comment but rather just a single isolated word. After this we will have to do some comparisons and for this, we will use the Levenshtein distance. The Levenshtein distance is one of the methods to calculate the similarity between two strings. It is calculated by the operation how many times the character is inserted, deleted or replaced when converting one string to the other.

We also need to have a confusion matrix which is a very crucial part of the whole technique because The purpose of this confusion matrix is to detect hidden words that social media users have masked using various symbols e.g., *b1tch*. The matrix will give us a probability between 0 and 1 and we need to incorporate that probability into the Levenshtein distance. Figure 1 shows an example of this matrix.

Lowercase Letter–Lowercase Letter	Uppercase Letter–Uppercase Letter	Uppercase Letter–Numeral (cont'd)
g and q	T and I	O and 0
p and n	D and O	B and 8
m and n	C and G	D and 0
y and z	L and I	S and 5
u and v	M and N	S and 8
c and e	P and B	Y and 5
cursive l and cursive b	F and R	Z and 7
cursive i and cursive e	U and O	T and 7
cursive a and cursive o	U and V	U and 0
Lowercase Letter–Numeral	E and F	U and 4
l and 1	V and W	**Numeral–Numeral**
b and 6	X and Y	0 and 8
o and O	cursive S and cursive L	3 and 9
g and 9	**Uppercase Letter–Numeral**	3 and 8
q and 9	G and 6	4 and 9
Uppercase Letter–Lowercase Letter	F and 7	5 and 8
I and l	Z and 2	5 and 3
	Q and 2	6 and 8
		7 and 1

Figure 1 – The most confused symbols [6]

References

1. *Sood S. O., Antin J., Churchill E. Using crowdsourcing to improve profanity detection // Proc. Of 2012 AAAI Spring Symposium Series. – 2012.*
2. *Indurthi V. et al. Identifying and Categorizing Offensive Language in Social Media using Sentence Embeddings // Proc. of SemEval@NAACL-HLT 2019. – 2019.*
3. *Pavlopoulos J. et al. Convai at semeval-2019 task 6: Offensive language identification and categorization with perspective and bert // Proceedings of the 13th international Workshop on Semantic Evaluation. – 2019. – P. 571-576.*
4. *Thenmozhi D. et al. SSN_NLP at SemEval-2019 Task 6: Offensive Language Identification in Social Media using Traditional and Deep Machine Learning Approaches //Proceedings of the 13th International Workshop on Semantic Evaluation. – 2019. – P. 739-744.*
5. *Pathak V. et al. KBCNMUJAL@ HASOC-Dravidian-CodeMix-FIRE2020: Using Machine Learning for Detection of Hate Speech and Offensive Code-Mixed Social Media text //arXiv preprint arXiv:2102.09866. – 2021.*
6. *Shastay A. Misidentification of Alphanumeric Symbols in Both Handwritten and Computer-Generated Information //Home healthcare now. – 2015. – Vol. 33. – №6. – P. 338-339.*