Automatic detection of hateful comments in online discussion

Hugo Lewi Hammer[1]

Oslo and Akershus University College of Applied Sciences, Norway,
hugo.hammer@hioa.no

Abstract. Making violent threats towards minorities like immigrants or homosexuals is increasingly common on the Internet. We present a method to automatically detect threats of violence using machine learning. A material of 24,840 sentences from YouTube was manually annotated as violent threats or not, and was used to train and test the machine learning model. Detecting threats of violence works quit well with an error of classifying a violent sentence as not violent of about 10% when the error of classifying a non-violent sentence as violent is adjusted to 5%. The best classification performance is achieved by including features that combine specially chosen important words and the distance between those in the sentence.

Keywords: *hateful comments,machine learning,threat detection*

1 Introduction

Over the past years there has been an alarming growth in hate against minorities and women [1, 2]. A similar increase in hate speech has been observed on the Internet [3, 4], and experts are concerned that individuals influenced by this web content may resort to violence as a result [5, 6].

Hateful speech is illegal by international law, signed and ratified by most countries in the world [7] (Article 20). In addition several countries, and most of the countries in the Western World, have similar national laws. There exist examples of individuals who have been arrested for expressing hate or threatening with violence, see e.g. [8, 9, 10, 11]. The last three examples are based on violent threats posted on the Internet. It is almost impossible for the police, NGOs, discussion moderators and others concerned with hate speech to keep track of the wast amount of online activities. A tool that automatically detects hate speech and threats of violence could potentially be very helpful.

The main aim of this paper is to evaluate the potential of using different machine learning approaches to detect sentences in hateful online discussions that contain a threat of or sympathy with violence (for short just called threats of violence in the rest of this introduction). More specifically this paper has the following contributions:

– A large text material of about 25.000 sentences from online discussion were manually annotated for threats of violence and about 1.500 were found. This large and unique material is openly available for further research.
– Reading through hateful discussions, one observes that much of the same words and phrases are used in different ways to express threats of violence. We present new approaches to construct features that capture these properties in an efficient way. The approaches are based on using the most salient words in violent sentences and constructing sentence features that are combinations of these words.

This paper is a major extension, both with respect to methodology and results, of the one page short paper [12].

2 Related work

There is little previous work specifically devoted to the detection of threats of violence in text. However, there is previous work which examines other types of closely related phenomena, such as cyberbullying and hate-speech.

Dinakar et al. (2011) [13] propose a method for the detection of cyberbullying by targeting combinations of profane or negative words, and words related to several predetermined sensitive topics. Their data set consists of over 50,000 YouTube comments taken from videos about controversial topics. The experiments reported accuracies from 0.63 to 0.80.

There has been quite a bit of work focused on the detection of threats in a data set of Dutch tweets [14, 15], which consists of a collection of 5000 threatening tweets. In addition, a large number of random tweets were collected for development and testing. The system relies on manually constructed recognition patterns in the form of n-grams, but details about the strategy used to construct these patterns are not given. In [15], a manually crafted shallow parser is added to the system. This improves results to a precision of 0.39 and a recall of 0.59.

Warner and Hirschberg (2012) [16] present a method for detecting hate speech in user-generated web text, which relies on machine learning in combination with template-based features. The task is approached as a word-sense disambiguation task, since the same words can be used in both hateful and non-hateful contexts. The features used in the classification were combinations of uni-, bi- and trigrams, part-of-speech-tags and Brown clusters. The best results were obtained using only unigram features, with a precision of 0.67 and a recall of 0.60. The authors suggest that deeper parsing could reveal significant phrase patterns.

3 Sentence features to detect threats of violence

Most classification methods within text mining are based on the so called document term matrix, also referred to as bag-of-words or unigram. A document term matrix counts word frequencies. For example the document term matrix for the two sentences:

Sentence 1: “I will kill Muslims and I will kill Jews”
Sentence 2: “We love to kill Muslims”

is shown in Table 1. For example we see that the word ’I’ occurs two times in ’sentence 1’ and zero times in ’sentence 2’. Further we see that the word ’kill’ occurs two times in ’sentence 1’ and once in ’sentence 2’. The columns of the document term matrix are referred to as *features* and represent the information from the sentences that will be used in the automatic text classification. Naturally there is information in the sentences that is not included in the document term matrix, e.g. the order of the words. Differences in the word order may change the meaning, e.g. the two sentences “I love Israel not Palestine,” and “I love Palestine not Israel” contain the same words, but differ in meaning. However, for the most typical application of text mining, like document summarization or clustering, the unigram matrix as introduced above works quite well. For the problem of detecting sentences with threats of violence, we hypothesize that classification can be improved by adding other specially constructed features as described below.

Table 1. Document term matrix for the two example sentences

	I	will	kill	and	Muslims	Jews	we	love	to
sentence 1	2	2	2	1	1	1	0	0	0
sentence 2	0	0	1	0	1	0	1	1	1

3.1 Bigrams of important words

We expect that a threat of violence often should contain the subject that wants to perform the violence, like ’I’ or ’we’, some aggressive words like ’kill’, ’bomb’, ’nuke’, ’gun’, etc, as well as the target for the violence, like ’Muslims’, ’Jews’, ’women’, ’bastards’, ’sandniggers’ and so on. Potentially important features from the sentences therefore are bigrams of such important words. A procedure to find important words is described below. If the important words in the sentences above are ’I’, ’we’, ’kill’, ’Muslims’ and ’Jews’, we start by excluding the non-important word from the sentences and then compute the bigrams of the remaining words. The result is shown i Table 2. For example we see that ’I-kill’ occurs two times in the first sentence and that ’kill-Muslims’ occurs in both sentences.

Table 2. Feature matrix for the bigram of important words.

	I-kill	kill-Muslims	Muslims-I	kill-Jews	we-kill
sentence 1	2	1	1	1	0
sentence 2	0	1	0	0	1

A natural extension of Table 2 is to use all combinations of the important words, not just the subsequent. The first features of 'sentence 1' will then be 'I-kill', 'I-Muslims', 'I-I', 'I-Jews', 'kill-Muslims', 'kill-I' and so on.

3.2 Bigrams of important words with distance function

Naturally we expect that a combination of important words like 'I-kill' is more important if 'I' and 'kill' are close to each other in the sentence, because then it is more likely that 'I' is related to 'kill'. Said in another way, if there are several words between 'I' and 'kill', it is more likely that 'kill' is used for another purpose in the sentence, and it is less likely that the sentence contains a threat of violence. To incorporate this in Table 2, we use a weight function. Maybe the most natural weight function is to divide by the number of words between the two important words in the sentence

$$w_1(d) = \frac{1}{d+1} \tag{1}$$

where d is the number of words between the two important words. If the two words are next to each other, $d = 0$. Using $w_1(d)$, Table 2 will be changed to Table 3. E.g. to compute 'I-kill' in 'sentence 1', we see that 'I-kill' occurs two times and both times with one word between, such that the computation becomes $1/(1+1) + 1/(1+1) = 1$. Further, 'we-kill' occurs once in 'sentence 2' with two words between, such that the computation becomes $1/(2+1) = 1/3$.

Table 3. Feature matrix for the bigram of important words using weight function (1).

	I-kill	kill-Muslims	Muslims-I	kill-Jews	we-kill
sentence 1	1	1	1/2	1	0
sentence 2	0	1	0	0	1/3

The selections of features above is based on using a set of important words. We chose those words that were significantly correlated with the response (violent/not-violent sentence).

4 Classification method

We base our classification of violent/non-violent sentences on logistic LASSO regression [17]. Logistic LASSO regression can document excellent classification properties, and is at least as effective as Support vector Machine [18]. A training set is used to estimate the parameters of the model. The fitted model can be used to compute the probability that a sentence (from a test set) contains a threat of violence or not. We classify a sentence using a threshold on the computed probability.

4.1 Evaluation of classification method

The first measure is the Mean Squared Error (MSE). Let $y_i \in \{0,1\}$ denote whether sentence $i \in \{1,2,\ldots,n\}$ contains a violent threat or not, where 1 means violent. Further let p_i denote the computed probability that sentence i is a violent sentence. The MSE is then computed as

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - p_i)^2 \tag{2}$$

We introduce one more measure. We denote the portion of non-violent sentences that are classified as violent as Type I error, and the portion of violent sentences that are classified as non-violent as Type II error. We adjust the threshold in the classification such that the Type I error is equal to α, say 0.05. Naturally we want the Type II error to be as small as possible under this restriction on the Type I error.

5 Evaluation

In this section we will compare the different suggested sentence features presented in Section 3 with respect to classification performance. Classification based on LASSO logistic regression model was performed using the 'glmnet' package [17] in the statistical software R [19].

5.1 Text material

The text material consisted of all comments on eight YouTube videos. All the videos were related to religious and political topics that typically create a lot of anger and disagreements in the comments, like the Eurabia theory [20], halal slaughter, Anders Behring Breivik, Geert Wilders, etc.

Each sentence in the material was manually annotated to either contained a threat or sympathy with violence or not. For sentences where it was impossible to decide, e.g. due to terribly poor language, or the sentence was part of a larger argument, the sentence were annotated as 'not violent'. A few comments contained copies of violent passages from the Bible or the Quran. Such sentences were classified as violent if the passage was violent. Sentences in other languages than English were removed from the text material. After the annotation process, the text material consisted of a total of 24,840 sentences where 1,469 sentences were violent. A randomly selected subset of 100 non-violent and 20 violent sentences (based on the annotation from the main annotator) were labeled by an other human annotator for inter-annotar study. The results showed that for 98% of the sentences, both annotators made the same decision, and both annotators found a total of 20 threats of violence.

5.2 Feature matrices

In this section we present the different feature matrices we want to compare, which are all based on the theory presented in Section 3. The first feature matrix is the unigram exemplified in Table 1. We denote this feature matrix UNI in the rest of this paper. In addition to the weight function w_1 in (1), we will also consider the weight function

$$w_2(d) = \exp\left\{-\left(\frac{d}{3}\right)^3\right\} \tag{3}$$

Comparing w_1 and w_2, w_2 gives more weight for small distances between important word and less weight for longer distances.

As previously described, we select important words using correlation with the response (in the training set) resulting in a list of approximately 300 important words depending on which part of the whole material were used as training set. The rest of the feature matrices consist of the features from UNI in addition to

- The feature matrix exemplified in Table 2; and together with UNI we denote this feature matrix C. ("C" (constant) since the score is independent of distance between the important words in the sentence)
- The same as C, but where we use all combinations of important words as described in the end of Section 3.1, denoted ACC (all combinations constant weight).
- The feature matrix exemplified in Table 3; and together with UNI we denote the feature matrix W1 (weight function w_1)
- The same as the item above, but with all combinations of important words, denoted ACW1.
- The same as W1, except that we use the weight function w_2. We denote this feature matrix W2.
- The same as the item above, but with all combinations of important words, denoted ACW2.
- We also select a random sample of 300 words (the same amount that we have of important words in the previous feature matrices) among the 3,000 most frequent words in the text material except stop words like *and, it, is, at, ...*. We compute the feature matrix based on these words the same way that we computed ACW1 and denote this feature matrix ACW1FW (frequent words). It is interesting to see how ACW1FW performs compared to ACW1 and the other feature matrices above.

Before computing the feature matrices above, all stop words and punctuation marks were removed from the sentences and all words were changed to lower case. In text mining it is also quite common to do word stemming, but for the application of detecting violent threats, different word stems are important, e.g. the word kill is more common in threats of violence than killed. Therefore word stemming was not performed. For computational robustness, we removed all features that were non-zero in three or fewer sentences.

5.3 Classification performance of the feature matrices

To evaluate the performance of the feature matrices we randomly divided the sentences into training and test sets. The training set consisted of 80% of the violent and 80% of non-violent sentences in the corpus, and the rest of the sentences constituted the test set. We fit the logistic LASSO regression model to the training set for each of the feature matrices in the previous section. We use the test set to evaluate the feature matrices by computing the Mean squared error (MSE) and Type II error measures as described in Section 4.1. For the Type II error, we set the Type I error to $\alpha = 0.05$. To reduce uncertainties in our results, we repeated this procedure 40 times. Tables 4 and 5 show the average MSE and Type II error over the 40 runs, respectively. For the tests in Table 4, we use paired Student's t-tests where the observations are the differences in MSE in

Table 4. The second column shows the average Mean squared error (MSE) for the different feature matrices over the 40 runs, computed using (2). The others show the results of tests for differences in MSE. The significance codes are adjusted for the number of tests carried out. That is: *** if p-value $\in [0, \frac{0.001}{28}]$, ** if p-value $\in (\frac{0.001}{28}, \frac{0.01}{28}]$, * if p-value $\in (\frac{0.01}{28}, \frac{0.05}{28}]$ and n.s. means not significant.

	MSE	ACW1	ACC	ACW2	W2	W1	C	UNI
ACW1	0.0269							
ACC	0.0277	**						
ACW2	0.0286	***	**					
W2	0.0290	***	**	n.s.				
W1	0.0290	***	***	n.s.	n.s.			
C	0.0294	***	***	***	**	**		
UNI	0.0325	***	***	***	***	***	***	
ACW1MF	0.0344	***	***	***	***	***	***	***

Table 5. The second column shows the average Type II error, in percentage, for the different feature matrices over the 40 runs. The other show the results of tests for the differences in Type II error. The significance codes are adjusted for the number of tests carried out. That is: *** if p-value $\in [0, \frac{0.001}{28}]$, ** if p-value $\in (\frac{0.001}{28}, \frac{0.01}{28}]$, * if p-value $\in (\frac{0.01}{28}, \frac{0.05}{28}]$ and n.s. means not significant.

	Type II	ACW1	ACC	ACW2	W1	W2	C	UNI
ACW1	9.72							
ACC	10.48	**						
ACW2	11.00	***	n.s.					
W1	11.04	***	n.s.	n.s.				
W2	11.11	***	n.s.	n.s.	n.s.			
C	11.16	***	n.s.	n.s.	n.s.	n.s.		
UNI	13.89	***	***	***	***	***	***	
ACW1MF	16.11	***	***	***	***	***	***	***

each of the 40 runs. For the tests in Table 5, we assume that the number of Type II errors for a feature matrix in one of the 40 runs is an outcome from a binomial distribution, and tests are performed approximating the binomial distributions with normal distributions. Since we perform several tests, the significance levels are adjusted using the Bonferroni correction [21].

From the tables we see that ACW1 performs significantly better than all the other feature matrices, both in terms of MSE and Type II error, e.g. being the only feature matrix with a Type II error below 10%. Further, the feature matrices using all combinations of the important words perform the best. We also see that the prediction performance significantly depends on the choice of weight function, with w_1 being the best. Finally we see that ACW1MF performs poorly showing that it is not enough just to include more features, but that those features must be cleverly chosen.

Table 6, shows contingency tables for the cases UNI and ACW1 based on the 40 runs.

Table 6. Contingency table of classification results.

UNI		**Truth**	
		violent	not violent
Prediction	violent	10127	8014
outcome	not violent	1633	151986
ACW1		violent	not violent
Prediction	violent	10617	8021
outcome	not violent	1143	151979

5.4 The strongest predictors of violent sentences

A great advantage of the LASSO model is that the solution is sparse with respect to features. To achieve the optimal predictions as summarized in Tables 4 and 5, the model consisted of between 400 and 1000 nonzero regression parameters (features). By choosing a higher value of the regularization parameter in the LASSO model, the solution will be even sparser, showing the few features which is the most important to detect threats of violence. The features is shown i Table 7. As expected, several of these strong predictors is a combination of a subject (I or we) and an aggressive word like kill, burn, die and so on. We also have word combinations that point to the target of the violent threat like "you-burn", "kill-you" and "kill-all". We also see that "breivik" is a strong predictor, finding sentences that support the terrible actions of Anders Behring Breivik. Lowering the value of the regularization parameter the set of nonzero features also include word combinations "breivik-hero" and "commander-breivik". It also included features that reduced the probability of a threat of violence like "not" and "never-kill".

Table 7. Features with a nonzero regression parameter for a high value of the regularization parameter.

kill-hell	kill	nuke	burn	and-burn	die
we-kill	you-burn	death-to	deported	deport	i-die
exterminate	kill-and	you-die	i-kill	i-burn	be-deported
breivik	shoot	kill-all	deport-and	kill-to	kill-you
and-die					

6 Closing remarks

In this article we have shown how text mining and machine learning can be used to detect threats of or sympathies with violence in online discussions. In particular we focus on how specially chosen features of two words can improve prediction compared to the traditional unigram (single words) feature matrix. Our results show that detecting treats of violence generally works quite well using machine learning with Type I and Type II errors of about 5 and 10%, respectively. Using all combinations of the important words and the weight functions w_1 gives the best results.

Parsing the text material may possibly improve our results, however, we expect that automatic parsers will have quite a hard time, since the language quality in such online discussions is terribly poor, full of slang, misprints and erroneous grammar.

One may also suggest to use ordinary bigrams, but then it is only possible to include features of word beside each other in the sentence. Including features of word separated in the sentence is impotant to detecting threats of violence. E.g. in Table 7 a feature like 'i-die' refers to sentences where the two important words are clearly separated in the sentence like 'I would love to see them die'. Also using ordinary bigrams, the number of features will be in the millions compared the only a few tens of thousands in our approach making the problem computationally less challenging and equivalent to ordinary unigram.

References

1. Fekete, L.: Pedlars of hate: The violent impact of the European far Right. Institute of Race Relations, http://www.irr.org.uk/wp-content/uploads/2012/06/PedlarsofHate.pdf (2013) [Online; accessed 12-March-2016].
2. Wilson, R., Hainsworth, P.: Far-right Parties and discourse in Europe: A challenge for our times. European network against racism, http://cms.horus.be/files/99935/MediaArchive/publications/20060_Publication_Far_right_EN_LR.pdf (2013) [Online; accessed 12-March-2016].
3. Goodwin, M., Ramalingam, V., Briggs, R.: The New Radical Right: Violent and Non-Violent Movements in Europe. Institute for Strategic Dialogue, http://www.strategicdialogue.org/ISD%20Far%20Right%20Feb2012.pdf (2013) [Online; accessed 12-March-2016].

4. Bartlett, J., Birdwell, J., Littler, M.: The rise of populism in Europe can be traced through online behaviour... Demos, http://www.demos.co.uk/files/Demos_OSIPOP_Book-web_03.pdf?1320601634 (2013) [Online; accessed 12-March-2016].
5. Strømmen, Ø.: The Dark Net. On Right-Wing Extremism, Counter-Jihadism and Terror in Europe. Cappelen Damm, Oslo, Norway (2012)
6. Sunde, I.M.: Preventing radicalization and violent extremism on the Internet (Norwegian). The Norwegian Police University College 2013:1 (2013)
7. UnitedNations: International Covenant on Civil and Political Rights. http://www.ohchr.org/en/professionalinterest/pages/ccpr.aspx (2014) [Online; accessed 12-March-2016].
8. TheTimesOfIndia: Akbaruddin Owaisi arrested in hate speech case. http://articles.timesofindia.indiatimes.com/2013-01-08/india/36216031_1_nirmal-rural-police-akbaruddin-owaisi-police-stations (2014) [Online; accessed 12-March-2016].
9. Skjetne, O.L., Hustadnes, H.: Ubaydullah Hussain is accused of violent threats (norwegian). http://www.dagbladet.no/2013/11/20/nyheter/ubaydullah_hussain/innenriks/islamisme/trusler/30429704/ (2014) [Online; accessed 12-March-2016].
10. Euronews: Neo-Nazi and black metal star Varg Vikernes arrested in France. http://www.euronews.com/2013/07/16/neo-nazi-and-black-metal-star-varg-vikernes-arrested-in-france-/ (2013) [Online; accessed 12-March-2016].
11. Valaker, O., Holte, M.A.: Bergen blogger arrested (norwegian). http://www.bt.no/nyheter/lokalt/Bergens-blogger-pagrepet-2732162.html (2012) [Online; accessed 12-March-2016].
12. Hammer, H.L.: Detecting threats of violence in online discussions using bigrams of important words. In: Intelligence and Security Informatics Conference (JISIC). (2014) 319–319
13. Dinakar, K., Reichart, R., Lieberman, H.: Modeling the detection of textual cyberbullying. In: The Social Mobile Web. (2011) 11–17
14. Oostdijk, N., van Halteren, H.: N-gram-based recognition of threatening tweets. In: Computational Linguistics and Intelligent Text Processing. Springer (2013) 183–196
15. Oostdijk, N., van Halteren, H.: Shallow parsing for recognizing threats in dutch tweets. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ACM (2013) 1034–1041
16. Warner, W., Hirschberg, J.: Detecting hate speech on the world wide web. In: Proceedings of the Second Workshop on Language in Social Media, Association for Computational Linguistics (2012) 19–26
17. Friedman, J., Hastie, T., Tibshirani, R.: Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software **33**(1) (2010) 1–22
18. Genkin, A., Lewis, D.D., Madigan, D.: Large-Scale Bayesian Logistic Regression for Text Categorization. Technometrics **49**(14) (2007) 291–304
19. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. (2013)
20. Fekete, L.: The Muslim conspiracy theory and the Oslo massacre. Technical Report 53(3): 30 – 47, Institute of Race Relations (2011)
21. Johnson, R., Wichern, D.: Applied Multivariate Statistical Analysis. Prentece Hall, Upper Saddle River, New Jersey (1998)